IR/RADAR SCENE ANALYSIS AND SPACE/TIME ALGORITHM ASSESSMENT

Research Associates for Defense Conversion, Inc.

M. Adel Slamani, Ph.D., and Braham Himed, Ph.D.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19960916 133

DTIC QUALITY INSPECTED 3

**Rome Laboratory
Air Force Materiel Command
Rome, New York**

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be releasable to the general public, including foreign nations.
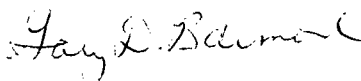
RL-TR- 96-121 has been reviewed and is approved for publication.

APPROVED:

DAVID D. FERRIS, JR.
Project Engineer

FOR THE COMMANDER:

GARY D. BARMORE, Major, USAF
Deputy Director of Surveillance & Photonics

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE July 1996	3. REPORT TYPE AND DATES COVERED Final Jun 94 – Mar 96

4. TITLE AND SUBTITLE
IR/RADAR SCENE ANALYSIS AND SPACE/TIME ALGORITHM
ASSESSMENT

6. AUTHOR(S)
M. Adel Slamani, Ph.D., and Braham Himed, Ph.D.

5. FUNDING NUMBERS
C – F30602-94-C-0203
PÉ – 61102F
PR – 2304
TA – E8
WU – PD

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research Associates for Defense Conversion, Inc.
10002 Hillside Terrace
Marcy NY 13403

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory/OCSM
26 Electronic Pky
Rome NY 13441-4514

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-96-121

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: David D. Ferris/OCSM/(315) 330-4432

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A procedure is designed for the Automated Statistical Characterization and Partitioning of Environments (A'SCAPE). A'SCAPE uses two procedures to partition regions into homogeneous areas. These are: (1) a mapping procedure that uses image processing to separate regions; and (2) a statistical procedure that uses statistics to separate regions. A'SCAPE is able to process real data through its image processing, statistical algorithms, and expert system rules. A parametric technique developed by LeCadre for the detection of signals in the presence of clutter and additive noise was also studied. This technique assumes an AR(MA) model of the interference. The order of the model is first estimated, the AR parameters are estimated, and then a covariance matrix is derived.

14. SUBJECT TERMS
Automated Statistical Characterization and Partitioning
Environments (A'SCAPE), Statistical algorithms

15. NUMBER OF PAGES
166

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. 239-18
298-102

DTIC QUALITY INSPECTED 3

Table of Contents

Part II: Performance Analysis of Multichannel Parametric

Detection Algorithms 85

Executive Summary

This report presents results from the basic contract and the three options. The report is divided into 2 parts. This approach was taken to organize the studies and investigations conducted and the results obtained into logical reporting categories. In the first part, we first review the approach to scene partitioning and statistical characterization originally developed at Syracuse University by Drs. M. A. Slamani and D. D. Weiner. The approach consisted previously of two main stages referred to as Mapping and Indexing. In the first stage, clutter patches are separated from background noise, and, in the second stage, clutter patches are separated from each other and the probability density function (PDF) is approximated for each patch.

The approach to scene partitioning and statistical characterization is remodeled so that it can be applied in general to separate any set of non-homogeneous patches without having to require that a background noise be present. Also, recognizing that two procedures are used to separate regions, the stages of the approach have been redesigned to reflect this fact. The two main stages of the approach are known as (1) the Mapping procedure, which uses image processing means to separate regions, and (2) the Statistical procedure, which uses statistical means to separate between them. The approach is assigned the acronym A'SCAPE which stands for the Automated Statistical Characterization And Partitioning of Environments. Though using two procedures, A'SCAPE needs four stages to achieve its aim at separating the different contiguous non-homogeneous regions that may exist in a scene. These stages consist of a preprocessing stage where classical time-frequency processing of the data is performed, a mapping procedure stage, a statistical procedure stage, and an indexing stage which assigns a set of descriptors for every pixel in the scene under investigation. When A'SCAPE is followed by a detection stage, all information is available with respect to which probability density function (PDF) should be used for each cell in the scene.

In order to enable A'SCAPE to process real data, the image processing and statistical algorithms had to be rewritten and new rules (in addition to the old ones) were selected to enable the convergence of the expert system which controls the flow between the different stages in A'SCAPE.

An example of a real IR scene is included in the report which consists of a lake region and a land region. First, the samples are selected in the preprocessing stage to result in a scene with uncorrelated data. Then, the mapping procedure separates the lake and land regions. Road ways and very small patches in the land region are successfully detected as outliers. Also, non-homogeneous regions (due mainly to trees) in the land region near the boundary between the land and lake are separated by the statistical procedure and grouped as homogeneous subpatches. Finally, every homogeneous declared region has its PDF approximated.

A demo package built in Matlab was also developed which describes in detail the different stages of A'SCAPE. The package has a friendly mouse driven graphical user interface (GUI) and consists of two main sections. The first section presents the detailed steps of the real IR data example. The second section is subdivided into two subsections where the first one consists of a set of examples which illustrate the need for A'SCAPE, whereas in the second a description is given for every stage of A'SCAPE. The views can be displayed manually or automatically. A movie is included that shows the steps through which the scene goes during the mapping procedure. This demo package is among the deliverables.

The objective of the second part was to study some specific parametric methods for the detection of signals in the presence of interferences such as noise, clutter and jamming. Specifically, we looked at a technique developed by LeCadre and studied its performance by comparing it to previously developed methods and against several parameters.

The scope of this effort was limited to the feasibility study of this method as far as its implementation is concerned. We investigated the performance of this technique and

studied all other effects which may contribute to its degradation and therefore reduce the probability of detection. LeCadre's technique is based upon assuming an AR or (ARMA) model of the interference. The first step in the technique is then to estimate the order of the model used. The second step involves estimating the AR parameters from which a covariance matrix is derived. Once this matrix is obtained, it will be used in conjunction with the detection problem to study two direction of arrival (DOA) techniques, namely MUSIC and ESPRIT, using the transformed or sometimes referred to as the "whitening" functional obtained from the inverse of the AR parameters covariance matrix.

Note that in this effort, we extended the ideas presented by LeCadre to include scenarios where signals, clutter and additive noise are included. These additions make this work very interesting since several improvements have been made to the original work. Moreover, the computer simulations of section 3 show that the results do match the theoretical development of section 2.

We feel that substantial additions to the beam-forming problem have been added in this effort and the results of the computer simulations describe the efficacy of the proposed algorithms.

Part I

Automated Statistical Characterization and Partitioning of Environments

with Application to IR Data

INTRODUCTION

1.1 - Problem Statement

In signal processing applications it is common to assume a Gaussian problem in the design of optimal signal processors. However, studies have shown that the Gaussian receiver performs very poorly in strong interference whenever the interference and signal spectra cannot be separated by filtering. For example, consider the spectra shown in Figure 1.1 consisting of 24 Doppler bins with uniformly spaced targets, indicated by the
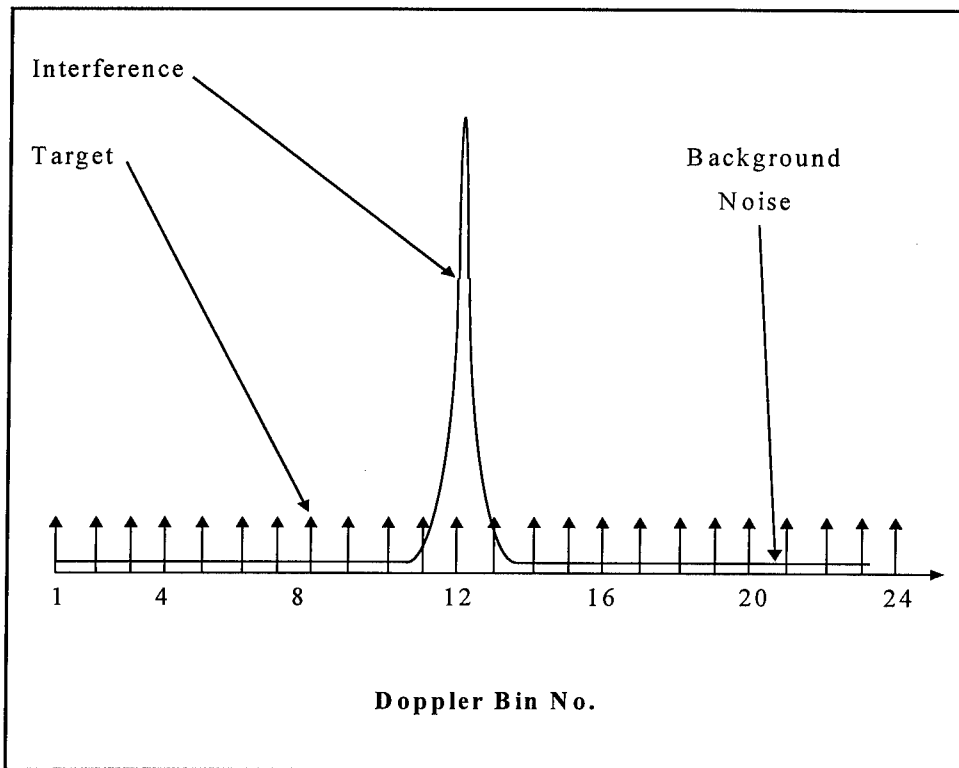
Figure 1.1 - Targets in Non-Gaussian Interference

small arrows, embedded in background noise and a bell shaped Gaussian interference. The optimal Gaussian based detector, referred to as the joint-domain localized generalized likelihood ratio receiver, is applied to each Doppler bin. The performance of

the receiver[1], shown in Figure 1.2, reveals that the probability of detection, PD, of the receiver is close to unity everywhere except for Doppler bins 11, 12, and 13 in which the strong Gaussian disturbance exists and PD falls rapidly to the probability of false alarm, PFA.

PD

1.0

0.8   Optimal Receiver

0.6

0.4   SNR = 0 dB

                                             INR = 50 dB

0.2   PFA = 10⁻⁵

         Center of
         clutter

0.0
   1   5   10   15   20   24

Doppler Bin Number
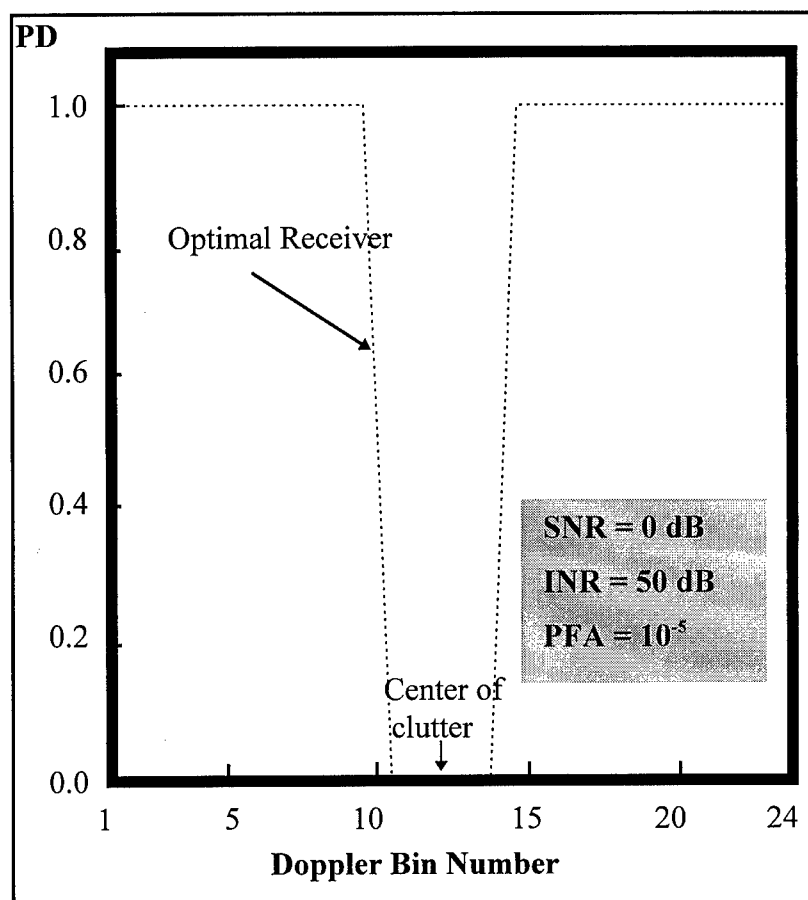
Figure 1.2 - <u>Performance of the Optimal Receiver</u>

Because of the fact that the interference is Gaussian and that the detector is based on the Gaussian distribution, we recognize that it is not possible to obtain any detection improvement. A question that arises is, "Could improved detection have been obtained in bins 11 to 13 had the disturbance been non-Gaussian ?". The answer is, "It is possible to achieve significant improvement in detection performance when the disturbance is non-Gaussian". To illustrate this answer, Table 1.1 presents the results of an example for a

[1] H. Wang, L. Cai, "On Adaptive Multiband Signal Detection with GLR algorithm." IEEE Trans. on AES, AES-27, no. 2, March 1991.

weak target in strong K-distributed clutter where the target and clutter spectra are coincident[2] (i.e., not separable). Note that the K-distributed based detector provides significantly improved performance even when the PD of the Gaussian detector approaches the PFA. For example, with SCR=0 dB and PFA=10^{-5}, PD equals 0.3 for the k-distributed detector and 10^{-5} for the Gaussian detector. This means that the K-distributed based detector can detect 3 out of 10 targets, whereas the Gaussian detector can detect only 1 in 100000 targets on the average. The important point is that detections can be made in a non-Gaussian environment using a non-Gaussian detector even when such performance is not possible with the Gaussian detector. Unfortunately, there are problems associated with non-Gaussian distributions: (1) there are a multitude of Non-Gaussian distributions, (2) some non-Gaussian distributions have shape parameters which result in different shaped probability density functions (PDFs), and (3) for a given set of data, it is difficult to determine which PDF can approximate the data.

SCR (dB)	PFA	PD (K-Distributed Detector)	PD (Gaussian Detector)
0	10^{-2}	0.50	0.06
-10	10^{-2}	0.40	0.02
-20	10^{-2}	0.22	0.01
0	$\mathbf{10^{-5}}$	**0.30**	$\mathbf{10^{-5}}$
-10	10^{-5}	0.25	10^{-5}
-20	10^{-5}	0.10	10^{-5}

Table 1.1 - Comparison of non-Gaussian and Gaussian based detectors

The use of PDF based detectors in the implementation of likelihood ratio tests, LRTs, and locally optimum detectors, LODs, for the general detection problem allows us

---

[2] P. Chakravathi, M. A. Slamani, D. D. Weiner, "Performance of the Locally Optimum Detector in a Correlated K-Distributed Disturbance," Proceedings of the 1993 National Radar Conference on Revolutionary Developments in Radar. Wakefield, MA, April 20-23, 1993.

to derive algorithms for performing both strong and weak signal detection, respectively, in a non-Gaussian environment. This is in contrast to classical detection which assumes a priori knowledge of the joint PDF underlying the received data and makes use of a single detector usually based on the Gaussian distribution.

Note that in practice, one cannot assume a priori knowledge of the joint PDF underlying the received data since the received data can come from a clear region, where background noise alone is present, or from a clutter region, where returns are due to reflections from such objects as ground, sea, buildings, birds,...etc. When a desired target return is from a clear region and the background noise is sufficiently small, the signal-to-noise ratio will be large and the strong signal detector (i.e, the LRT) should be used. However, if a desired target return is from a clutter region, two situations can exist. When the desired target can be separated from the clutter by means of space-time processing and the background noise is sufficiently small, the signal-to-noise ratio will be large and a strong signal detector should again be used. When the desired target cannot be separated from the clutter by means of space-time processing and the clutter return is much larger then the desired target return, then a weak signal detector (e.g., LOD) based on the PDF of the clutter should be used. Use of the LOD in a strong signal situation can result in a severe loss in performance. Hence, it is necessary for the receiver to determine whether a strong or weak signal situation exists.

Figure 1.3 summarizes the different cases which may arise depending on whether the target is embedded in background noise or in background noise plus clutter. This can result either in a strong signal case, intermediate signal case or a weak signal case. Note that the Gaussian assumption is used for the cases where the likelihood-ratio-test (LRT) and generalized likelihood-ratio-test (GLRT) detectors are utilized. For the weak signal case, the PDF of the region has to be approximated in order to enable the use of the appropriate LRT or the sub-optimal locally-optimum-detector (LOD).

All of this suggests the necessity for a procedure to 1) continuously monitor the environment, 2) subdivide the surveillance volume into homogeneous patches, and 3) select the appropriate detector for processing the data. In addition, depending on

statistical changes in the environment over time and space, the process enables the receiver to adapt so as to obtain close to optimal performance. This is achieved by the Automated Statistical Characterization and Partitioning of Environments (A'SCAPE) procedure, previously used successfully on simulated radar data[34].
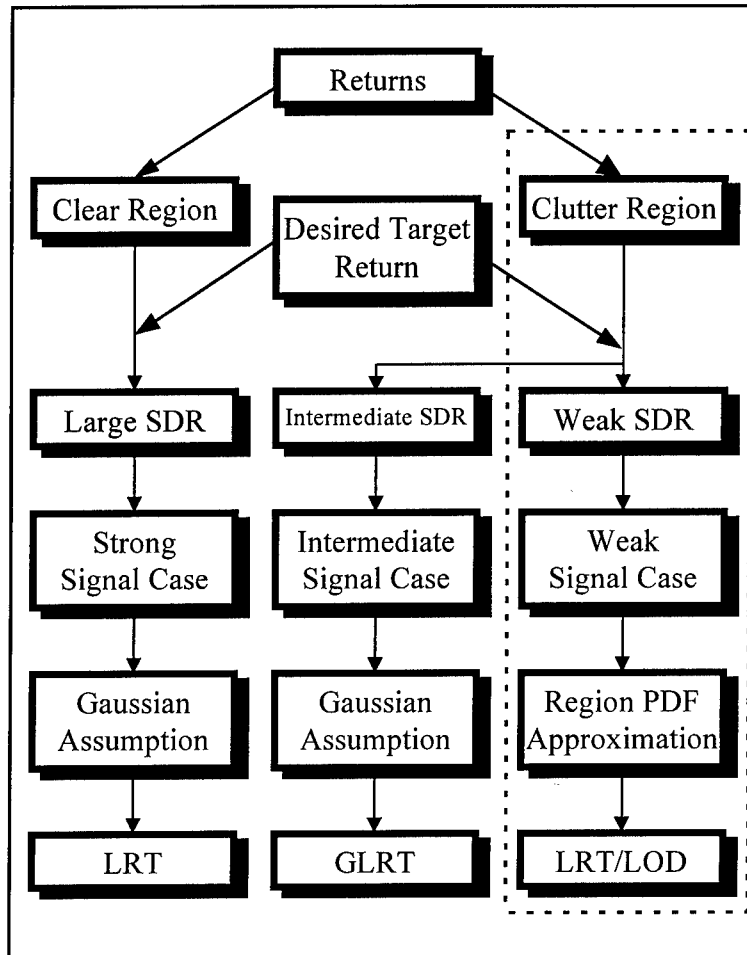
Figure 1.3 - Different Target Cases

---

[3] M. A. Slamani and D. D. Weiner, "Rationale Behind the Use of Image Processing to Partition a Radar Surveillance Volume into Background Noise and Patches," Proceedings of the 1993 36th Midwest Symposium on Circuits and Systems.

[4] M. A. Slamani and D. D. Weiner, "Use of Image Processing to Partition a Radar Surveillance Volume into Background Noise and Patches," Proceedings of the 1993 Conference on Information Sciences and Systems.

A'SCAPE uses two separate procedures to determine all homogeneous regions and sub-regions in the scene. The first procedure, referred to as the mapping procedure, is used to separate contiguous homogeneous regions by segregating between their power levels. The second procedure, referred to as the statistical procedure, separates contiguous homogeneous regions by segregating between their data distributions. The statistical procedure uses the Ozturk algorithm, a newly developed algorithm for analyzing random data[5]. Furthermore, the statistical procedure identifies suitable approximations to the probability density function for each region. Convergence of the mapping and statistical procedures are controlled through expert system rules as developed in Chapter 4.

1.2 - A'SCAPE Procedure

A'SCAPE is based on the approach to scene partitioning and statistical characterization originally developed at Syracuse University by Drs. M. Adel Slamani and D. D. Weiner. The approach consisted previously of two main stages referred to as Mapping and Indexing. In the first stage, clutter patches are separated from background noise, and, in the second stage, contiguous non-homogeneous clutter patches are separated from each other and the probability density function (PDF) is approximated for each patch.

The approach is remodeled so that it can be applied in general to separate any set of contiguous non-homogeneous patches without having to require that a background noise be present. Also, recognizing that two procedures are used to separate regions, the stages of the approach have been redesigned to reflect this fact. The two main stages of the approach are known as (1) the Mapping procedure, which uses image processing means to separate between regions, and (2) the Statistical procedure, which uses statistical means to separate between regions. The approach is assigned the acronym A'SCAPE which stands for the Automated Statistical Characterization And Partitioning of Environments. Though using two procedures, A'SCAPE needs four blocks to achieve its goal of separating the different contiguous non-homogeneous regions that may exist in

---

[5] A. Ozturk and E. J. Dudewicz, "A New Statistical Goodness of Fit Test Based on Graphical Representation," The Biomedical Journal, 1991.

a scene. As shown in Figure 1.4, the first block is a preprocessing block that performs classical space-time processing on the collected data (e.g., inphase and quadrature components extraction, envelope detection). The second block separates contiguous homogeneous patches and subpatches by segregating between their average power levels based on the mapping procedure described in chapter 2. The next block, presented in chapter 3, goes one step further and separates homogeneous subpatches by segregating between their probabilistic data distributions. Furthermore, this block identifies suitable approximations to the probability density function (PDF) of each homogeneous patch and determines the location of outliers in the scene. The final block, labeled indexing, indexes the scene under investigation by assigning a set of descriptors to every pixel in the scene. For each pixel, the indexing is used to indicate to which homogeneous patch the pixel belongs, whether it is an edge pixel, whether it is an outlier, which pixels can be chosen as reference pixels if the pixel is to be tested, and which PDF best approximates the data in the pixel. Note that the reference pixels are pixels which belong to the same homogeneous patch and which are closest to the pixel to be tested.
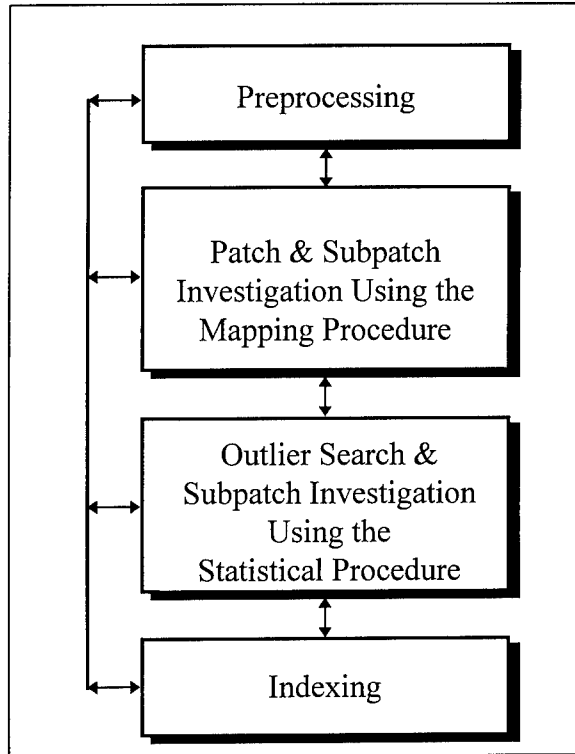
Figure 1.4 - A'SCAPE Block Diagram

The forward and backward interactions between the different blocks are controlled through rules of an expert system shell referred to as Integrated Processing and Understanding of Signals (IPUS) developed jointly by the University of Massachusetts and Boston University[6]. IPUS is discussed in chapter 4.

Note that when A'SCAPE is followed by a detection stage (e.g., target detection in Radar), all information needed is available for every pixel in the scene. Furthermore, given the PDF that can approximate the patch in which the test pixel is located, the appropriate detector is readily selected. This is in contrast to the classical detection approach where a single detector (usually the Gaussian detector) is used in processing the entire scene.

The mapping and statistical procedures are presented in chapters 2 and 3, respectively. The expert system shell is discussed in chapter 4. An example illustrating the different stages of A'SCAPE when applied to real data of an IR image is given in chapter 5.

---

[6] H. Nawab, V. Lesser, et al., Integrated Processing and Understanding of Signals in Symbolic and Knowledge-Based Signal Processing, Edited by A.V. Oppenheim and H. Nawab, Prentice Hal, Ennglewood RPsiffs, N.J., 1991.

2

Mapping Procedure

2.1 - Introduction

In this chapter, a mapping procedure is presented to partition a scene into homogeneous regions by segregating between the power levels of the different patches.

Given a scene which consists of a set of non-homogeneous patches, the mapping procedure starts first by separating the regions into a patch with the lowest power level, referred to as the lowest patch (LP), and remaining patches (RPs). The procedure is then repeated to isolate the next patch with the lowest power level (LP) and so on until it is not possible to find anymore LPs. General observations are first made on LP and RPs.

2.1.1 - Observations on LP and RPs

Assume that a scene consists of JxK pixels and that JxK data magnitudes P(j,k) are available to the mapping procedure. In this case, JxK pixels need to be mapped into LP and RPs. Let's, examine the nature of the LP and RPs in order to understand the theory behind the procedure developed for mapping.

The following observations are based on computer generated examples of LP and RPs data where the RPs-to-LP (RLR) ratio is assumed to be greater than 0 dB.

2.1.1.a - Observations on LP

- On average, the LP data values are smaller than the RPs data values.

- Large data values exist in a LP that may be higher than some data values of the RPs.

- Large data values in the LP tend to be isolated points.

- The number of LP data significantly larger than the average is relatively small. Figure 2.1 shows a typical LP data histogram.

- The relatively small number of LP pixels with large data values are distributed evenly throughout the LP.
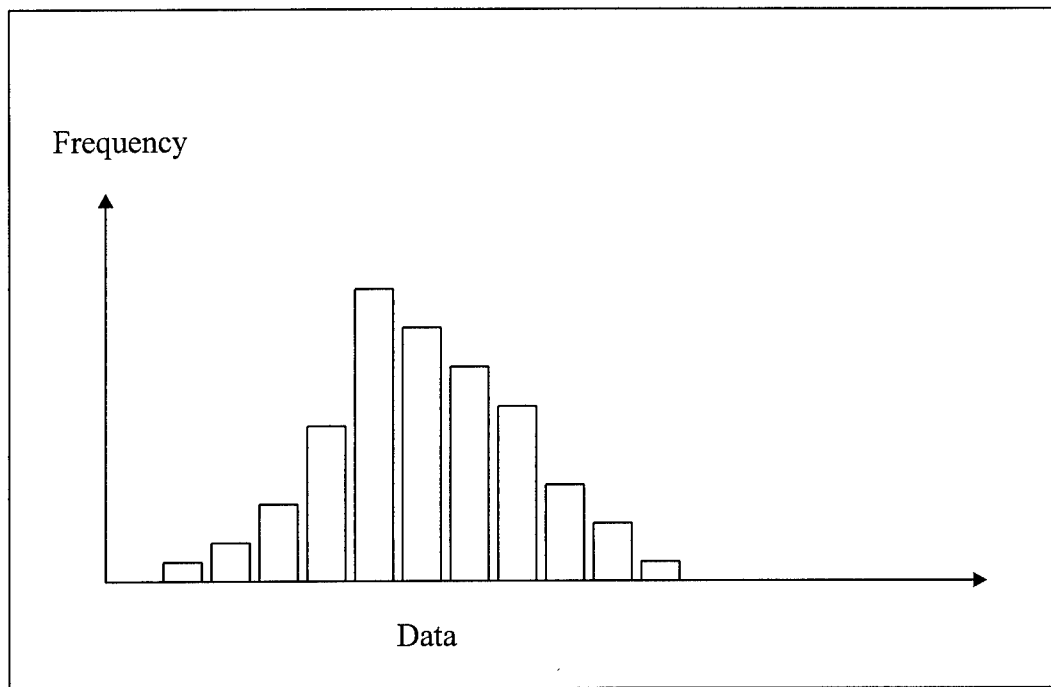
Figure 2.1 - <u>Typical LP data histogram</u>

2.1.1.b - <u>Observations on RPs</u>

- On average, RPs data values are higher than LP data values.

- The large RPs data values are larger than the largest LP data values assuming positive RPs-to-LP-Ratio (RLR).

- Small RPs data values exist and may be smaller than the large LP data values.

- Large data values in the RPs tend to be clustered.

2.2 - <u>Mapping Procedure</u>

Using the fact that the RPs patches, on average, have stronger magnitudes, the mapping procedure begins by setting a threshold that results in a specified fraction of LP pixels. Image processing tools based on data thresholding and region partitioning are then used to establish the LP and RPs patches. If the final image contains a significantly different fraction of LP than originally established by the initial threshold, the process is repeated with a new threshold. The mapping procedure iterates until it is satisfied that the final scene is consistent with the latest specified threshold. Then, edges of all patches are

detected using an image processing technique based on the unsharp masking[78]. Finally, all regions in the scene are assigned a unique identifying number.

The mapping procedure is therefore composed of three steps. The first step results on the identification of LP and RPs, whereas the second and third steps consist of the enhancement and detection of patch edges and labeling of the different regions, respectively. These three steps are explained next.

2.2.1 - Identification of LP and RPs

Identification of LP and RPs is performed by the following steps: thresholding, quantization, corrections and assessment.

2.2.1.a - Thresholding and Quantization

Identification of LP and RPs starts by setting a threshold q that results in a specified fraction of LP declared pixels. Then, a quantized volume is formed as follows: all pixels with magnitude less than q are given a value of 0 (zero) and all pixels with magnitude above q are given a value of 1 (one). Let Q(j,k) represent the quantized value of the jkth pixel. Then,

$$Q(j,k) = \begin{cases} 1 & if \ P(j,k) \geq q \\ 0 & if \ P(j,k) < q \end{cases} \quad j=1,2,...,J \text{ and } k=1,2,...,K \quad (2.1)$$

where P(j,k) is the magnitude of the jkth pixel.

2.2.1.b - Corrections

Consider a set of 3x3 pixels. As shown in Figure 2.2, let the center pixel be referred to as the test pixel and the surrounding pixels be referred to as the neighboring

---

[7] R. Gonzalez, P. Wintz, "Digital Image Processing." 2nd edition, Addison-Wisley Publishing Company, Nov. 1987.

[8] E. Hall, Computer Image Processing and Recognition. Academic Press, 1979.

12

pixels. Assume that a patch cannot be formed by a single pixel. In this case, every test pixel in a patch has at least one neighboring pixel that belongs to the same patch.

A test pixel belonging to a RPs patch that has at least one neighboring LP pixel is referred to as a RPs edge pixel (RPsE). On the other hand, a test pixel that belongs to a RPs patch for which none of the neighboring pixels are in the LP is referred to as an inner RPs pixel. Vice versa, a test pixel belonging to a LP patch that has at least one neighboring RPs pixel is referred to as a LP edge pixel (LPE). Also, a test pixel that belongs to a LP patch for which none of the neighboring pixels are in the RPs is referred to as an inner LP pixel.
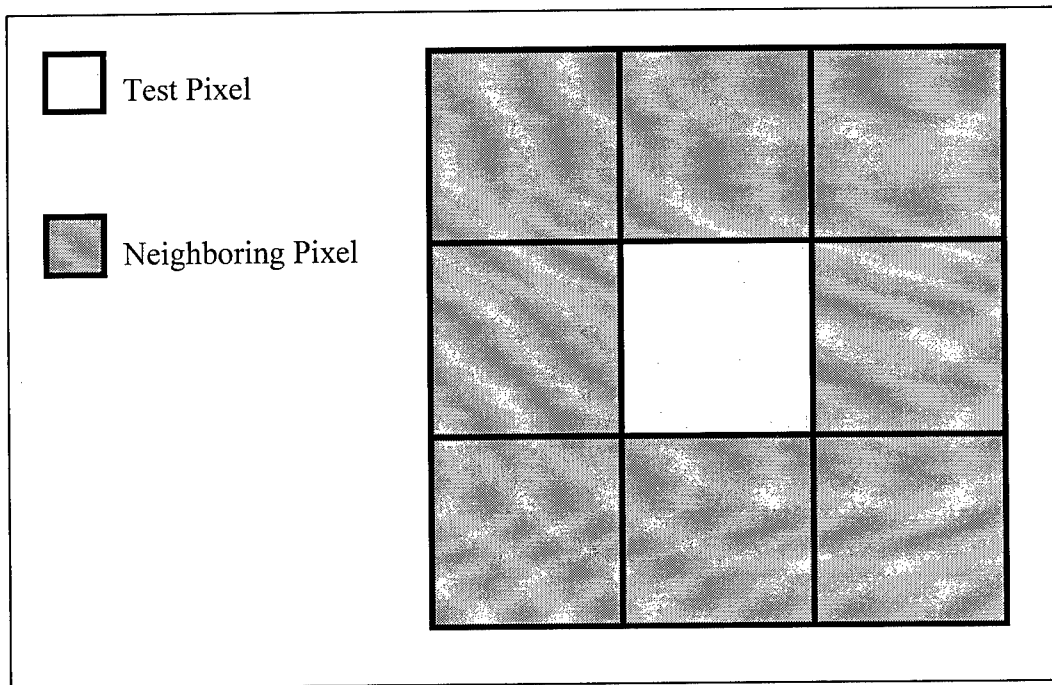
Figure 2.2 - 3x3 pixels

The proposed correction technique consists of transforming the quantized volume into a "corrected" volume. The transformation consists of the following steps:

- **1.** In the quantized volume, declare as LP pixels all pixels with quantized values $Q(j,k)=0$ and as RPs pixels all pixels with quantized values $Q(j,k)=1$.

- **2.** Choose the necessary number of RPs neighboring pixels, NCQ, for a test pixel in the quantized volume to be declared as a RPs pixel in the corrected volume. NCQ can take one of the following values: 5,6,7,8.

- **3.** For every test pixel in the quantized volume count the number of neighboring RPs pixels. If the number is greater than or equal to NCQ declare the test pixel as a RPs pixel in the corrected volume. Otherwise, declare the test pixel as a LP pixel in the corrected volume.

When all the pixels of the quantized volume have been tested, a "corrected" volume consisting of declared LP or RPs pixels is obtained.

Because NCQ is chosen to be relatively large (i.e. NCQ=5,6,7 or 8), LP pixels that were incorrectly identified in the quantized volume as RPs pixels due to their large power tend to be reclassified as LP pixels. Also, inner RPs pixels in the quantized volume are recognized as RPs pixels in the "corrected" volume. Meanwhile, most of the RPs edge pixels in the quantized volume are recognized as LP pixels in the "corrected" volume. This results in an over-correction where most of the RPs edge pixels are identified as LP pixels. As an example, when NCQ=8, only inner RPs pixels in the quantized volume are recognized as RPs pixels in the "corrected" volume and all RPs edge pixels in the quantized volume are recognized as LP pixels in the "corrected" volume. In order to recover the edge pixels, a second correction stage is needed where the "first corrected" volume will be transformed into a "second corrected" volume. Let the "first corrected" volume be referred to as the "corrected-quantized" volume (CQV) and the "second corrected" volume be referred to as the "corrected-corrected" volume (CCV). The following steps are used to transform the CQV into the CCV:

- **1.** Choose the necessary number of RPs neighboring pixels, NCC, for a test pixel in the CQV to be declared as a RPs pixel in the CCV. NCC can take one of the following values: 1,2,3 or 4.

- **2.** For every test pixel in the CQV count the number of neighboring RPs pixels. If the number is greater than or equal to NCC, declare the test pixel as a RPs pixel in the CCV. Otherwise, declare the test pixel as a LP in the CCV.

14

2.2.1.c - <u>Assessment</u>

Let LPQP, LPCQP and LPCCP denote the percentage of LP pixels in the quantized, "corrected-quantized" and "corrected-corrected" volumes, respectively. LPQP is pre-specified so as to determine the threshold for the quantized volume, whereas LPCQP and LPCCP are computed after the CQV and the CCV are obtained.

The assessment process consists of comparing LPCQP and LPCCP to LPQP in order to determine whether or not the percentages of the LP pixels after correction are consistent with the percentage of LP pixels in the quantized volume. When there is no consistency, further quantization, correction and assessment are performed until consistency is obtained.

The thresholding/quantization, first correction, second correction, and assessment stages are used to find the best threshold to separate between LP and RPs patches. Once LPQP has been set a threshold is computed. Then, corrections are made to try and build the LP region and the RPs patches . The correction stages re-label some of the above-threshold pixels as LP pixels if they are likely to belong to the LP, and some of the below-threshold pixels as RPs pixels if they are likely to belong to a RPs patch, based on the choices for NCQ and NCC. Depending on the quality of the threshold choice many or few pixels are re-labeled. At the end of the procedure, LPCCP is computed and compared to LPQP, if the values are within a certain range, few pixels would have been re-labeled, the threshold is accepted and the assessment passes. Otherwise, many pixels would have been re-labeled and the threshold is rejected. The iterative process continues then by setting another threshold through the choice of a new value for LPQP.

Rules for choosing NCQ, NCC and LPQP and for determining when consistency of the percentages is obtained are explained in Chapter 4.

2.2.2 - <u>Enhancement and Detection of Patch Edges</u>

Once the LP and RPs have been detected, The mapping procedure enhances the patches and detects their edges. These are done by the smoothing, edge detection and edge enhancement stages presented next.

2.2.2.a - <u>Smoothing</u>

In many cases of simulated data, examples have shown that when the percentages are consistent, declared patches may contain isolated LP declared pixels. Because small magnitudes can arise in a RPs patch as explained in the Section 2.1.1, it is most likely that the LP isolated pixels in the RPs patches are RPs pixels. The smoothing process is used to detect these isolated pixels and label them adequately by transforming the CCV into a smoothed volume (SV). The smoothing technique consists of the following steps:

- **1.** Choose the necessary number of RPs neighboring pixels NS for a LP identified test pixel in the CCV to be declared as a RPs pixel in the SV where NS can take one of the following values: 5,6,7, or 8.

- **2.** For every LP identified pixel in the CCV count the number of neighboring RPs pixels. If the number is greater than or equal to NS, declare the test pixel as a RPs pixel in the SV. Otherwise declare the test pixel as a LP pixel in the SV.

2.2.2.b - <u>Detection of Patch Edges and Edge Enhancement</u>

(i) - <u>Detection of Patch Edges</u>

After smoothing, each pixel in the SV has been declared as either a LP or a RPs pixel. The next step is to determine which of the RPs pixels are located on the edges of the RPs patches and which of the LP pixels are located on the edges of the LP patch. This is important for subsequent signal processing if reference pixels for estimating parameters of a test pixel are to be chosen properly.

Identification of RPs edge pixels (RPsE) and LP edge pixels (LPE) is done by the use of an image processing technique referred to in the image processing literature as the unsharp masking. It consists of the following steps:

<u>To detect RPsE pixels</u>

1 - A weighting filter consisting of a 3x3 array of pixels is constructed, as shown in Figure 2.3, where the center pixel has a weight given by $w(0,0)=8$ and the neighboring pixels have weights given by $w(-1,-1)=w(0,-1)=w(1,-1)=w(-1,0)=w(1,0)=w(-1,1)=w(0,1)=w(1,1)=-1$. The center pixel is positioned on the test pixel. Notice that the weights of the filter pixels sum to zero. In particular,

16

w(-1,-1)=-1	w(0,-1)=-1	w(1,-1)=-1
w(-1,0)=-1	w(0,0)=8	w(1,0)=-1
w(-1,1)=-1	w(0,1)=-1	w(1,1)=-1

Figure 2.3 - <u>Weighting Filter</u>

SQ(j-1,k-1)	SQ(j,k-1)	SQ(j+1,k-1)
SQ(j-1,k)	SQ(j,k)	SQ(j+1,k)
SQ(j-1,k+1)	SQ(j,k+1)	SQ(j+1,k+1)

Figure 2.4 - <u>Quantized values of the 3x3 array corresponding to the jkth pixel</u>

$$\sum_{m=-1}^{1} \sum_{n=-1}^{1} w(m,n) = 0 \qquad (2.2)$$

2 - Assume that the weighting filter is centered at the jkth pixel in SV. The pixels corresponding to the 3x3 array of the weighting filter have quantized values as illustrated in Figure 2.4. By definition,

$$SQ(j,k) = \begin{cases} 1 & \textit{if the jkth cell in SV is declared as RPs} \\ 0 & \textit{if the jkth cell in SV is declared as LP} \end{cases} \qquad (2.3)$$

where j=1,2,...,J and k=1,2,...,K.

To avoid filter pixels falling outside SV, the coordinates of the jkth pixel at which the filter is centered are constrained to j=2,3,...,J-1, and k=2,3,...,K-1.

3 - Form the sum,

$$S = \sum_{m=-1}^{1} \sum_{n=-1}^{1} w(m,n) SQ(j+m,k+n) \qquad (2.4)$$

- If S is equal to zero, all pixels have the same assigned value. This can arise only when the test pixel is not an edge pixel.

- If S is positive, the test pixel is an edge pixel and is labeled as such.

- If S is negative, the test pixel cannot be an edge pixel. On the other hand, one or more of the neighboring pixels are guaranteed to be an edge pixel.

The three situations are illustrated in Figures 2.5-a, b, c and d. In Figures 2.5-a and 2.5-b, S=0 because all 9 pixels are in LP and RPs, respectively. Observe that the test pixel is not an edge pixel. In Figure 2.5-c, S=4>0. Note that the test pixel is an edge pixel. Finally, in Figure 2.5-d, S=-2<0 and the test pixel is not an edge pixel.

To detect LPE pixels,

First, Equation 2.3 is rewritten as follows,

$$SQ(j,k) = \begin{cases} 0 & \textit{if the jkth cell in SV is declared as RPs} \\ 1 & \textit{if the jkth cell in SV is declared as LP} \end{cases} \qquad (2.5)$$

where j=1,2,...,J and k=1,2,...,K.

18
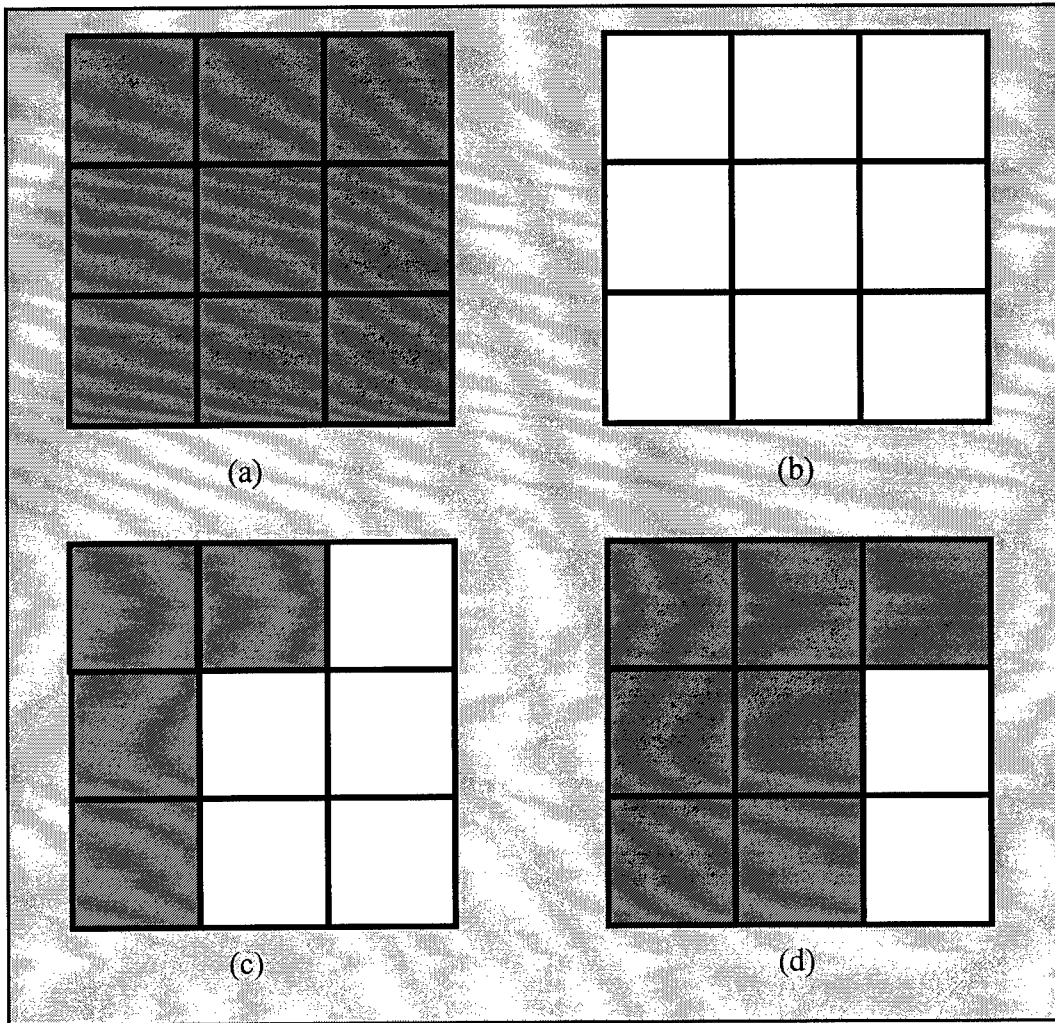
Figure 2.5 - <u>Example of Unsharp Masking Technique</u>

(a) S=0, (b) S=0, (c) S>0, (d) S<0

Then, step 3 above is repeated.

At the end of the edge detection procedure, each pixel in the original volume has been labeled as either RPs, LP, RPsE, or LPE.

(ii) <u>Enhancement of Patch Edges</u>

The edges of the smoothed volume (SV) tend not to follow the irregular edges that may actually exist. Consequently, the edges are further enhanced by examining the magnitudes of pixels just outside the LPE pixels and the RPsE pixels. If the magnitudes

of these pixels exceed the last threshold q for which the assessment passes, they are declared as RPsE pixels otherwise they are declared as LPE pixels.

At the end of the edge enhancement procedure, edges are detected and each pixel in the original volume is labeled as either LP, RPs, LPE, or RPsE pixel.

2.2.3 - <u>Labeling of Patches</u>

When the first round of the mapping procedure is completed, recall that the mapped volume has a value of 0 assigned to LP pixels and a value of 1 assigned to the RPs pixels. Therefore, nothing more needs to be done for the LP region as all of its pixels are already indexed by the number 0. On the other hand, all pixels in each of the RPs patches are assigned a value of 1. Thus, a numbering procedure has to be implemented to enable the computer to distinguish between the various RPs patches. The approach taken in this work is to assign every pixel in the first patch investigated the number 1, every pixel in the second patch investigated the number 2, and so on until all patches have been indexed with consecutive integers. In this way, all pixels in each RPs patch are assigned a unique number.

If a pixel belongs to a new RPs patch, the key to the numbering is the ability to recognize this fact. This is done by considering a mask of 5 pixels as shown in Figure 2.6 where the white pixels represent neighboring pixels and the shaded one is the test pixel to be numbered.

If $M(j,k)$ is the value assigned to the ijth pixel in the mapped volume, then

$$M(j,k) = \begin{cases} 0 & \textit{if the jkth pixel is declared as LP} \\ 1 & \textit{if the jkth pixel is declared as RPs} \end{cases} \qquad (2.6)$$

Assuming that the test pixel to be numbered is the jkth pixel in the original surveillance volume, let the assigned number be denoted by $N(j,k)$. Also, let G denote the unique number assigned to the RPs patch previously investigated and H the minimum positive number assigned to the neighboring pixels. Then, by definition, we have that,
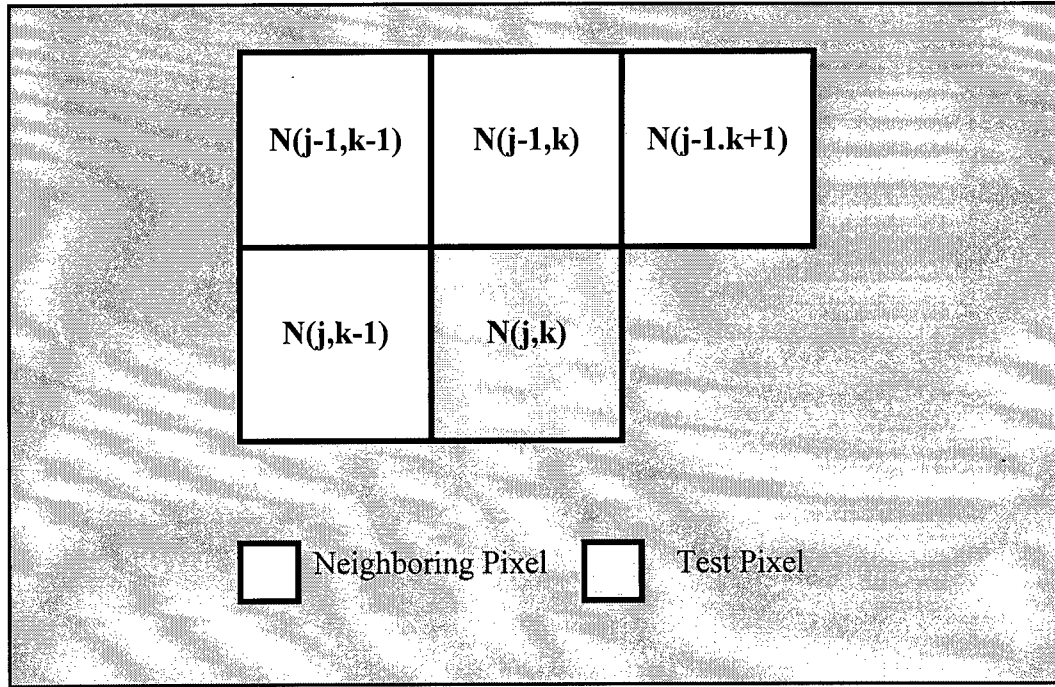
Figure 2.6 - <u>Mask used in numbering</u>

$$N(j,k) = \begin{cases} 0 & \text{If } M(j,k) = 0 \\ G+1 & \text{If all neighboring cells are numbered } 0 \\ H & \text{If at least one of the neighboring cells is numbered nonzero} \end{cases} \quad (2.6)$$

The number G is incremented by unity whenever a new RPs patch is detected.

Because a RPs patch boundary may be sharply shaped, as shown in the example of Figure 2.7, the numbering procedure may end up by assigning two different numbers for different pixels of the same RPs patch. This anomaly is avoided by further testing the neighboring pixels of the pixel to be numbered as follows:

1 - For a given pixel to be numbered, look up the numbers assigned to the set of neighboring pixels (j-1,k-1), (j,k-1), (j+1,k-1), and (j-1,k),

2 - Take the minimum nonzero number of those in step 1,

3 - Reassign all nonzero numbered neighboring pixels the minimum nonzero number from step 2,

21

4 - Revisit all the pixels in the surveillance volume that have been numbered so far. If any pixel is assigned a nonzero number identical to one of those in step 1, reassign that pixel the minimum nonzero number of step 2.

For example, with respect to Figure 2.7, the above steps have the effect of assigning a value of 1 to all pixels of the RPs patch shown.

Once numbering is completed, the LP pixels are assigned a value of 0, and every RPs patch is assigned a unique positive number.
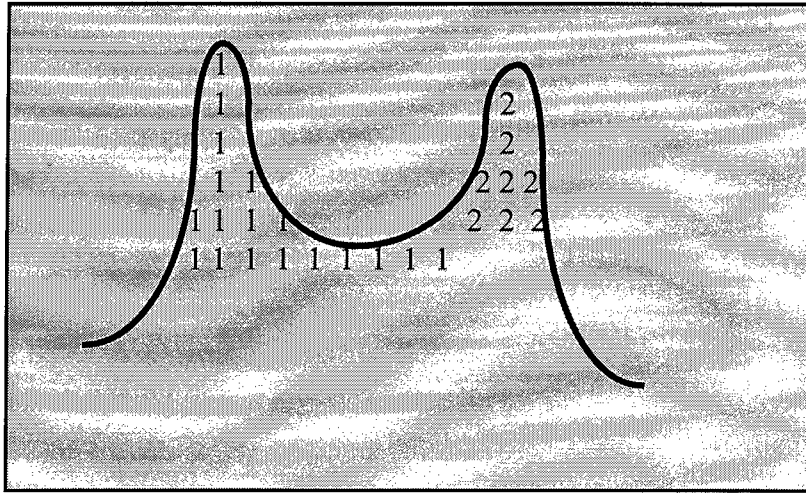
Figure 2.7 - <u>Example of a sharply shaped boundary</u>

2.3 - <u>Conclusion</u>

The mapping procedure consists of the following steps: thresholding/quantization, correction, assessment, smoothing, edge enhancement, edge detection, and labeling of patches. As explained in the previous sections and shown in Figure 2.8, these are subdivided into three main stages referred to as the Identification of LP and RPs, the detection of patch edges, and labeling of patches, respectively.
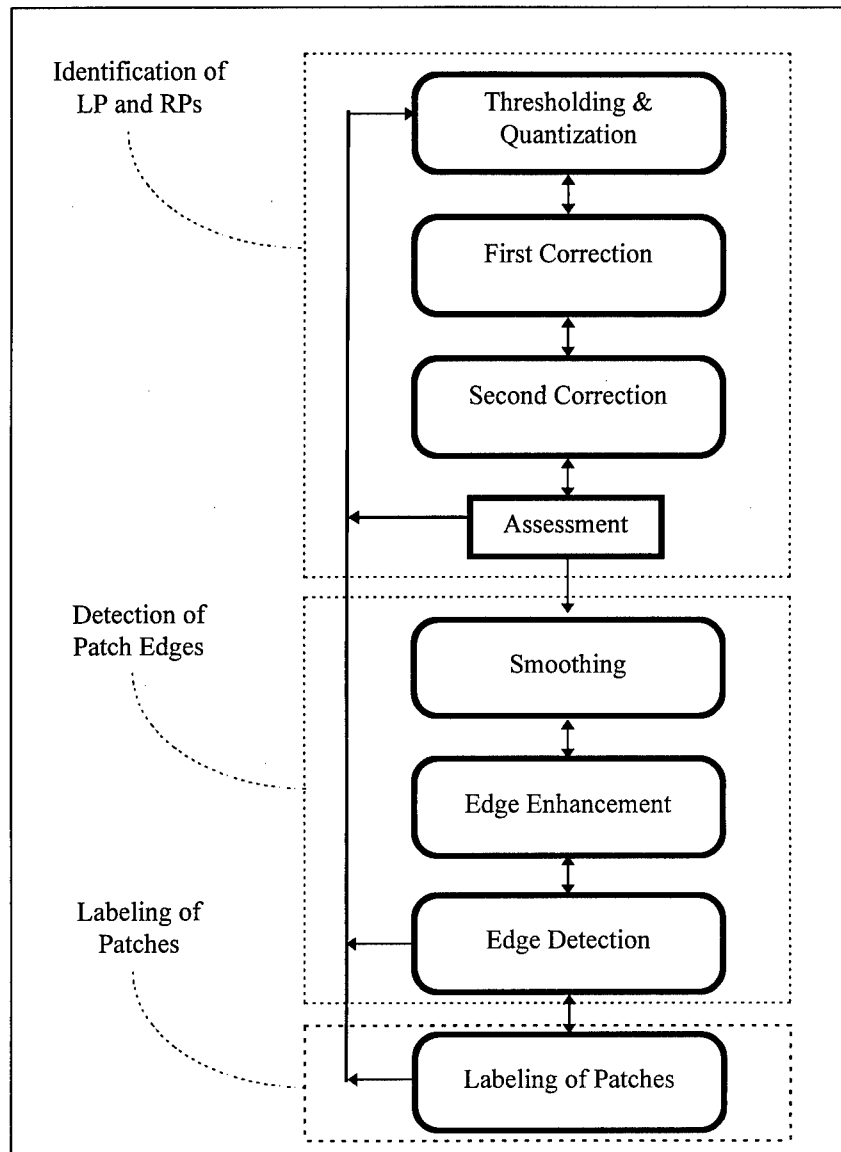
Figure 2.8 - <u>Block Diagram of the Mapping Procedure</u>

As stated before, once numbering is completed, the LP pixels are assigned a value of 0, and every RPs patch is assigned a unique positive number.

Recall that mapping consists of appropriately selecting a threshold to distinguish between LP and RPs patches using only the assumption that the LP magnitudes, on average, are smaller than the RPs magnitudes. This same approach may be used once again to extract a LP subpatch from a set of contiguous RPs subpatches of higher magnitudes in a given RPs patch. In this case, the RPs patch containing non-

homogeneous RPs subpatches will be viewed as a volume containing a RPs subpatch region with low power level and a set of subpatches with higher power levels occupying the rest of the RPs patch area.

The mapping procedure is therefore used to extract the RPs subpatch with the lowest power (which becomes an LP) from among the remaining RPs subpatches in a given RPs patch. Because the numbering stage has already labeled each patch with a unique number it is straight forward for the program to select a patch and check for the presence of subpatches in it.

For each RPs patch, the mapping procedure is performed iteratively until it is hypothesized that every subpatch in a given RPs patch is homogeneous and cannot be partitioned further. After all RPs subpatches have been extracted, the surveillance volume will consist of a set of homogeneous patches with different power levels.

3

Statistical Procedure

3.1 - Introduction

In Chapter 2, a mapping procedure was presented to partition a scene into homogeneous regions by segregating between the power levels of the different patches. When no more patches (sub-patches) can be found, the mapping procedure ends and is followed by the statistical procedure. The statistical procedure is applied to every homogeneous previously declared patch or subpatch to (1) further separate non-homogeneous subpatches having very similar power levels but different statistical distributions, (2) locate outliers in the scene, and (3) approximate the probability density function (PDF) of each homogeneous region.

The Ozturk algorithm is used by the statistical procedure to approximate the PDF of each patch and is presented next followed by the definition of outliers. The strategy used in the statistical procedure is discussed in the last subsection.

3.2 - Introduction to the Ozturk Algorithm

The Ozturk algorithm is based on sample order statistics and is used for univariate distribution approximation. This algorithm has two modes of operation. In the first mode, the algorithm performs a goodness of fit test. The test determines, to a desired confidence level, whether the random data is statistically consistent with a specified probability distribution. In the second mode of operation, the algorithm approximates the PDF underlying the random data. In particular, by analyzing the random data and without any a priori knowledge, the algorithm identifies from a stored library of PDFs that density function which best approximates the data. Estimates of the location, scale, and shape parameters of the PDF are provided by the algorithm. The algorithm has been found to work reasonably well for observation sizes as small as 100. Throughout this work, 100 reference pixels are selected for approximating the PDF of each test pixel. Note that when a region contains less than 100 pixels it is discarded and is not processed by the statistical procedure stage.

3.2.1 - <u>Goodness of Fit Test</u>

The goodness of fit test determines whether or not the set of data samples provided to the algorithm is statistically consistent with a specified distribution, referred to as the null hypothesis. Let NR denote the number of reference pixels. For the null hypothesis, the program utilizes a Monte-Carlo simulation of 2,000 trials to generate an averaged set of NR linked vectors in the UV plane, as shown in Figure 3.1. Using the standardized sample order statistics of the data, the program then creates a second system of NR linked vectors in the UV plane. The terminal points of the linked vectors, as well as the shapes of their trajectories, are used in determining whether or not to accept the null hypothesis. The null hypothesis is the distribution against which the sample data is to be tested.
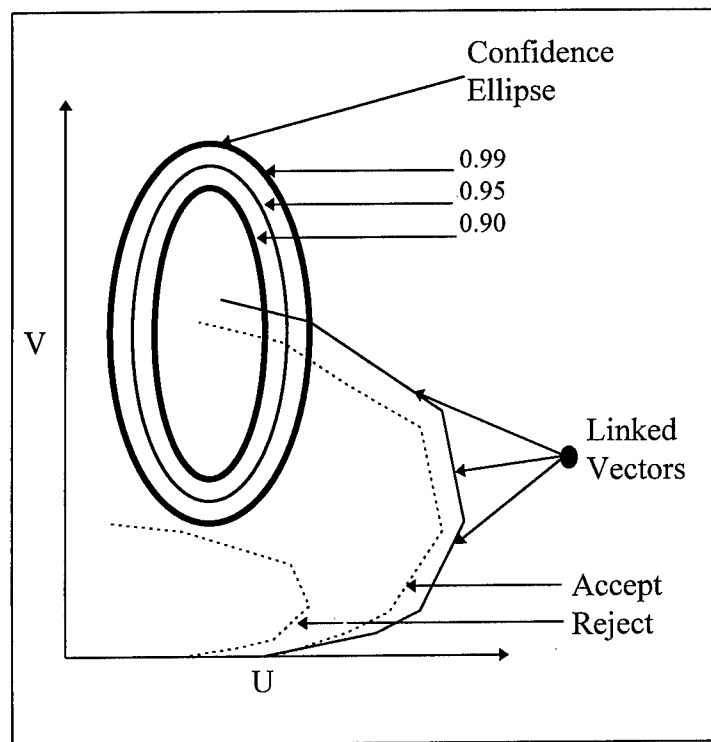
Figure 3.1 - <u>Goodness of fit test</u>

The algorithm provides quantitative information as to how consistent the sample data set is with the null hypothesis distribution by use of confidence contours where each contour is derived from a specified probability that the end point falls within the contour

26

given that the data comes from the null distribution. An example of these contours is shown in Figure 3.1 for probabilities of 0.9, 0.95, and 0.99. If the end point of the sample data linked vector locus falls within a contour, then the sample data set is said to be statistically consistent with the null hypothesis at a confidence level based on the probability specified for that contour. If the sample data set is truly consistent with the null hypothesis, the system of sample linked vectors is likely to closely follow that for the system of null linked vectors.

3.2.2 - Approximation Chart Mode

The approximation chart mode is simply an extension of the goodness of fit test mode. Following a similar approach to that outlined in the section for the goodness of fit mode, random samples are generated from a library of different univariate probability distributions. In the goodness of fit test mode, the locus end point was obtained for the null hypothesis and sample size, NR. For the approximation chart mode we go one step further by obtaining the locus end point for each distribution from the library of distributions for the given sample size, NR, and for various choices of the shape parameter(s). Thus, depending on whether it has a shape parameter or not, each distribution is represented by a trajectory or point in the two dimensional UV plane. Figure 3.2 shows an example of the approximation chart. Note that every point in the approximation chart corresponds to a specific distribution. That point closest to the sample data locus end point is chosen as the best approximation to the PDF underlying the random data. This closest point is determined by projecting the sample locus end point to all points on the approximation chart and selecting that point whose perpendicular distance from the sample point is the smallest. Once the PDF underlying the sample data is selected, the shape, location and scale parameters are then approximated.

3.3 - Outliers

Outliers that may exist in a set of reference pixels would seriously change the statistical distribution of the set of data under examination. Outliers can cause a problem
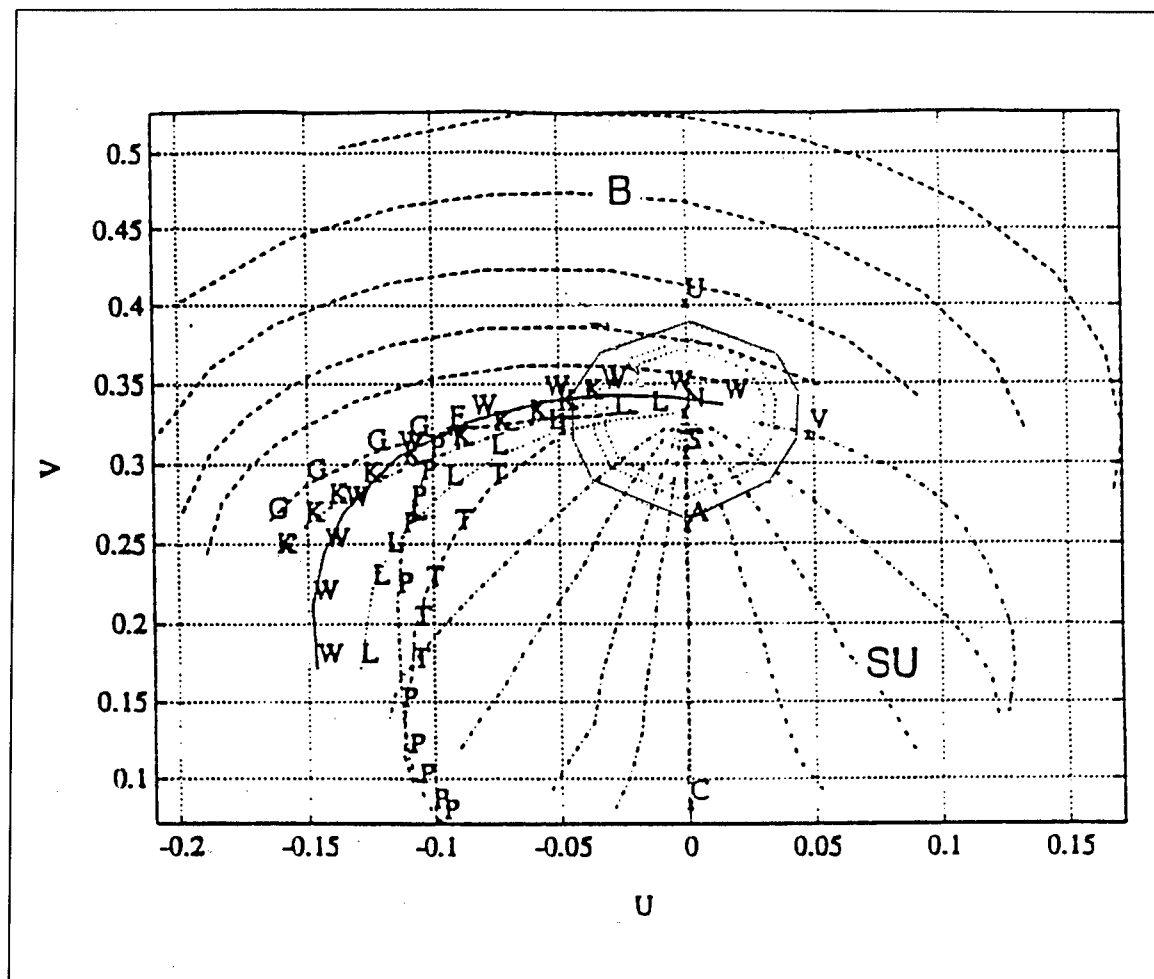
Figure 3.2 - <u>Approximation Chart, N=Normal, U=Uniform, E=Negative Exponential, A=Laplace, S=Logistic, C=Cauchy, V=Extreme Value, T=Gumbel type-2, G=Gamma, P=Pareto, W=Weibull, L=Lognormal, K=K-distributed, B=Beta, and SU=Su-Johnson</u>

in correctly approximating the PDF underlying a set of data by significantly altering the set of linked vectors generated by the Ozturk algorithm. This has been illustrated through an example in[9] where a set of NR=100 reference data, referred to as set A, were generated from the Lognormal distribution with shape parameter 0.01. Also, another set, referred to as set B, was formed which contained 97 data points from set A and three data points

---

[9] M. A. Slamani, "A New Approach to Radar Detection Based on the Partitioning and Statistical Characterization of the Surveillance Volume," Ph.D., Syracuse University, December 1994.

28

with very small values to constitute the outliers in the set. The two sets were processed by the Ozturk algorithm and have their locus end points (indicated by the arrow) reproduced in the approximation charts of Figures 3.3 and 3.4. Note how the end

Figure 3.3 - <u>Approximation chart for the end point of set A</u>

point in Figure 3.4 for the set containing outliers (set B) is far removed from the Lognormal PDF from which 97 out of the 100 data points of set B were generated. Investigating the cause, it was noted that the linked vectors in set B are smaller than those of set A causing the locus end point for set B to fall way below the end point for set A and, therefore, outside the confidence ellipses. This is due to the fact that the amplitudes of the linked vectors are proportional to the magnitude of the standarized data which depend on the mean and standard deviation of the set. The three outliers do not affect

significantly the mean of the set but do increase the variance tremendously causing the standardized data and, therefore, the amplitudes of the linked vectors to become smaller. In that example, the mean and standard deviation for set A are equal to 32.54 and 4.69, respectively, while the mean and standard deviation for set B are equal to 31.63 and 18.08, respectively. This example illustrates what can happen when three LP pixels with small data values are misidentified and associated with a set of RPs pixels.

Figure 3.4 - <u>Approximation chart for the end point of set B</u>

Outliers may be due to (1) misidentified LP pixels in the RPs or vise-versa, (2) pixels having data values of very low probability of occurrence, (3) pixels with non-representative data, and (4) pixels containing signals from strong targets. One way to identify outliers within a region composed of a set of 100 reference pixels is to compute

30

the mean m and the standard deviation s within that region and designate as an outlier any pixel whose data value falls outside the interval [m-ks,m+ks] where k is an empirical parameter (usually between 1.5 and 3) to be set by the user. In our case k is set equal to 2.

3.4 - Test pixel selection

In order to approximate the PDF of a given RPs patch, a set of test pixels is chosen in that RPs patch. Also, 100 reference pixels are identified for each test pixel that belong to the selected RPs patch and are the closest to the test pixel in order to obtain the approximating PDF for each test pixel. This will provide information on how the data is distributed in the selected RPs. Thus it is important to know how to select the test pixels and their corresponding reference pixels. Test and reference pixels selection involves three steps:

1. A RPs patch is chosen from among the declared RPs patches. This can be done automatically by the program since at this stage every RPs patch has been labeled with a unique number.

2. A set of NT test pixels is then chosen in the RPs patch being processed where the value of NT depends upon the extent to which the patch needs to be characterized. Note that any pixel in the RPs patch can be a test pixel. The best choice for the test pixels is when they are evenly spread throughout the entire area of the RPs patch.

3. Finally, for each test pixel, a set of reference pixels is selected. The reference pixels must be in the same RPs patch as the test pixel and should be the closest to it because of the assumption that the reference pixels are representative of the test pixel.

In order to select the reference pixels for a given test pixel, the program starts by centering a 3x3 mask around the test pixel and choosing as reference pixels those neighboring pixels within the mask that are declared to be in the same RPs patch as the test pixel. If the desired number of reference pixels are not obtained, the program increases the size of the mask by adding one row and one column to each boundary of the 3x3 mask. This results in a 5x5 mask where only the pixels in the augmented rows and columns need to be examined. The process of adding one row and one column to each

31

boundary of the previous mask continues until the desired number of reference pixels have been obtained.

3.5 - Strategy to Sub-Patch Investigation Using the Statistical Procedure

The statistical procedure is applied to every patch and subpatch that has been declared as being homogeneous by the mapping procedure. For each patch or subpatch, a set of test pixels evenly spread throughout the patch and their 100 closest reference pixels are first selected. Let each set of 100 pixels be referred to as a tile. As explained in the previous Section, note that the sets of 100 reference pixels are chosen to be disjoint, the closest to and belonging to the same patch as their respective test pixels. As shown in the block diagram of Figure 3.5, the statistical procedure consists of four steps that are performed as follows:

1. Using the goodness of fit test of the Ozturk algorithm, a Gaussianity check is performed on every tile to ensure whether the data in the tile are Gaussian or not. This results on every patch having its tiles labeled as either Gaussian or non-Gaussian.
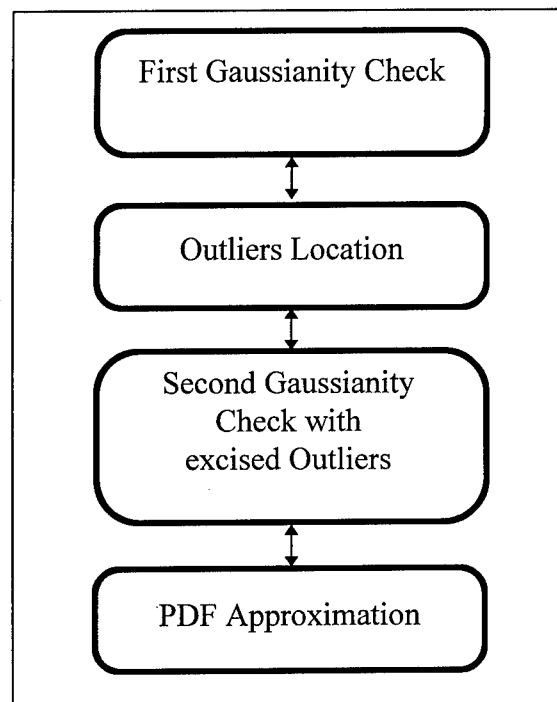

Figure 3.5 - Statistical Procedure

2. Existing outliers are located in those tiles declared as non-Gaussian.

3. For every non-Gaussian declared tile, pixels with outliers are excised from the tile and replaced with the closest pixels to the tile whose data are not outliers. Next, the Gaussianity check is performed once again as in step (1). At the end of this step, each patch will have tiles declared as Gaussian or non-Gaussian and pixels with outliers.

4. Using the Ozturk algorithm, the (U,V) coordinates of the locus end point is obtained for every tile declared as non-Gaussian in step (3). Next, a check is made to ensure whether or not the data of the set of tiles which constitutes a sub-patch can fit within a confidence ellipse. This is done by first declaring a subpatch every set of contiguous non-Gaussian tiles. Then, computing the average (Uav,Vav) coordinates of all test pixels of the same subpatch and getting its best approximating PDF and the corresponding confidence ellipse. Finally, a check is made whether all (U,V) coordinates of the test pixels are within the confidence ellipse of the average coordinates (Uav,Vav). If not, the tiles are regrouped so that all (U,V) coordinates for each group of tiles can fit within the same ellipse as their corresponding average (Uav,Vav) coordinates. Each group forms then a subpatch.

When the statistical procedure ends, each pixel in every patch is declared as Gaussian, non-Gaussian, or outlier. In addition, existing subpatches formed by the sets of contiguous tiles, whose (U,V) coordinates fit under the same confidence ellipse as their corresponding average (Uav,Vav) coordinates, are detected.

4

Expert System Shell IPUS

4.1 - Introduction

The expert system shell used to control the different stages of the mapping and indexing procedures is the Integrated Processing and Understanding of Signals (IPUS) developed jointly by the University of Massachussets and Boston University. IPUS architecture utilizes the fact that signal processing theories supply the system designer with a signal processing algorithm (SPA) that has adjustable control parameters. Each instance corresponding to a particular set of fixed values for the control parameters is referred to as an SPA instance. The IPUS architecture is designed to search for appropriate SPA instances to be used in order to accurately model the unknown environment. The search is initiated by detection of a discrepancy at the output of a given SPA due to the fact that the signal being monitored by the SPA does not satisfy the requirements of the SPA instance. Once a discrepancy has been detected, a diagnosis procedure is used to identify the source of the distortion that may have led to the discrepancy. Then, either parameters of the same SPA can be readjusted or a different SPA can be chosen to reprocess the data.

In our case of interest, each block in the different stages of the A'SCAPE processor consists of an SPA and SPA instances. Rules have been developed which enable the detection of discrepancies at the output of the SPAs, and identification of different possible distortion sources that would cause the discrepancies. Note that the arrows connecting different blocks in Figures 1.4, 2.8, and 3.5 are bi-directional. This is to allow for a system to assess its decisions, correct any discrepancies, and adapt to any changes in the environment being monitored.

4.2 - IPUS

The IPUS architecture has evolved from research on the design of a sound

34

understanding system[10]. The goal of such a system is to identify the origins of various sound sources (such as telephones, vacuum cleaners, crying infants, etc.). The complexity of the sound understanding problem arises because of two factors:

1. The need to process a tremendous variety of signal types due to the situation dependent nature of the input. For example, not only may the input of a sound understanding system include different types of signals, such as narrow-band, impulsive, harmonic signals, but may also include various combinations of these signals.

2. The need to change processing goals in a context dependent way. For example, the goal of a signal understanding system might be to respond to either the sounds of an infant or a ringing telephone and to ignore other sound sources. If an infant sound is detected, the system would then ignore the telephone and would switch its main goal to determining whether the infant is laughing, crying or choking.

These two factors also arise in the image processing problem addressed in this report. Specifically, complexity is encountered because of:

1. The need to process a tremendous variety of signal types due to the situation dependent nature of the input. For example, the PDFs of the random data in the pixels may be Gaussian, Weibull, K-distributed, etc., with various values for the scale, location, and shape parameters.

2. The need to change processing goals in a context dependent way. For example, the usual operational mode in imaging is the enhancement and segmentation of regions. If a region has suspicious characteristics (e.g., isolated tiny region that may be a target, or signs of a tumor in medical imaging) the operational mode should change into isolating the area and investigating its nature.

---

[10] H. Nawab, V. Lesser, et al., Integrated Processing and Understanding of Signals in Symbolic and Knowledge-Based Signal Processing, Edited by A.V. Oppenheim and H. Nawab, Prentice Hal, Ennglewood RPsiffs, N.J., 1991.

The algorithms employed in the IPUS architecture to identify the mathematical models for all operations of A'SCAPE (e.g., partitioning, detecting edges, approximating PDF) are referred to as signal processing algorithms (SPA's).

Because of the two factors mentioned previously, it is very difficult, or even impossible, to design a single mathematically derived signal processing algorithm that can be applied to all possible input signals to produce the desired information for each input. To deal with such complexities, the approach taken in the IPUS architecture is for the signal understanding system to have access to a "data-base" of mathematically derived algorithms. For the imaging problem, examples are partitioning algorithm, edge enhancement and detection algorithms, goodness of fit test algorithm, and the PDF approximation algorithm. This data base is indexed by the type of assumptions made about the input signal and the type of output information desired in accordance with the current goals of the signal understanding system.

For example, it may be assumed that the region under investigation is Gaussian. A goodness of fit test algorithm would be applied to determine whether the data is statistically consistent with the Gaussian assumption. If the Gaussian assumption is not rejected, then the desired output information would be the location and scale parameters.

The IPUS architecture utilizes the fact that signal processing theories often supply a system designer with a signal processing algorithm that has adjustable control parameters (sampling period of data samples, number and location of reference pixels, etc.). SPA denotes a data base of SPA "instances", each instance corresponding to a particular set of fixed values for the control parameters. The IPUS architecture is designed to search for appropriate SPA instances to be utilized in particular situations in order to accurately model the unknown environment.

Two basic approaches for carrying out the signal processing are:

1. Process the incoming signal with all the SPA-instances that are potentially relevant to the entire class of possible input signals in the application domain and then choose the output data that has the most consistent interpretation. This approach requires vast

36

amounts of signal-processing output data to be examined by the higher level interpretation processes.

2. Process the incoming signal with one or a small number of the possibly relevant SPA instances, then use some mechanism to recognize if incorrect processing has taken place. This is followed by determining the nature of the incorrect processing through a diagnostic reasoning process, and finally changing the parameter settings of the SPA with the goal of obtaining an SPA-instance which is appropriate for the processing of the input signal. The SPA-instance with adjusted control parameter settings is then used to reprocess the input signal.

In order to select appropriate values for the SPA control parameters, the system must consider the current system goals as well as knowledge about certain characteristics of the particular input signal. This leads to the dilemma that choosing the appropriate control parameter values requires knowledge about the signal, but this knowledge can only be obtained by first processing the signal with an algorithm with appropriate control parameter setting. The IPUS architecture uses an iterative technique for converging to the appropriate control parameter values. The technique begins by using the best available guess for the SPA control parameters values. The SPA instance output is then analyzed through a discrepancy detection mechanism for indicating the presence of distorted SPA output data. A diagnosis is then performed for mapping the detected discrepancies to distortion hypotheses. A signal reprocessing phase then proposes a new set of values for the control parameters of the SPA with the goal of eliminating the hypothesized distortions. The SPA instance corresponding to the new control parameter values is then used to reprocess the input signal. The output from the reprocessing once again undergoes discrepancy detection and if necessary is followed by diagnosis, signal reprocessing planning, and further reprocessing of the input signal.

The signal data and the interpretation hypotheses derived from that data are stored on a blackboard with hierarchically organized information levels. The hypotheses on the blackboard fall into two basic categories: hypotheses posted to explain the signal data and hypotheses posted to specify expectations about the nature of the signal data. The

inferencing on the blackboard is performed by different knowledge sources (KS's) for tasks such as discrepancy detection, diagnosis and reprocessing, and data interpretation. These tasks are presented in the following sections.

4.2.1 - Discrepancy Detection

Ideally, application of an SPA instance to input data results in undistorted output data. However, if the control parameters of the SPA instance are not appropriately chosen, distorted output data may result. The key to discrepancy detection is the ability to recognize and Classify discrepancies due to distortion introduced by the SPA instance. Three types of discrepancies are possible:

1 - The first type of discrepancy is referred to as a <u>violation</u>. A violation occurs when the SPA output data implies the presence of a signal that is not a member of the allowable Class of input signals. For example, disturbances arising from pixels in a clear region are always modeled as Gaussian processes because of the expectation that background noise is Gaussian. Suppose that the output data from an SPA instance implies that the disturbance from a pixel in the Clear region is non-Gaussian. This constitutes a violation type of discrepancy.

2 - The second type of discrepancy is referred to as a <u>conflict</u>. A conflict occurs when the current SPA output data is inconsistent with expectations arising from interpretations of past data. There are two types of conflicts depending upon whether all, or only a portion, of the current SPA output data is inconsistent. For an example of the first type of conflict, suppose previous SPA output data arose from disturbances in the clear region while current SPA output data is arising from disturbances in a patch. A conflict of the first type occurs if all of the current SPA output data, such as an increase in variance and non-Gaussianity of the data, conflict with previous interpretations from the clear region. For an example of the second type of conflict, suppose that previous SPA output data has resulted in the interpretation that the disturbance is from the Clear region. This might be implied by the SPA output data indicating Gaussian statistics, zero mean, and a variance level in the range of the background noise. A conflict of the second type occurs when, even though Gaussian statistics are confirmed by the current SPA output

38

data, they also reveal that the mean is no longer zero and the variance level has increased significantly. This could happen if the disturbance is now coming from a Gaussian patch where the data is highly correlated.

3 - The third type of discrepancy is referred to as a <u>fault</u>. A fault can also arise in two different ways. The first kind occurs when two or more different SPA's that are applied to the same data result in different output interpretations. The other kind occurs when two or more instances of a single SPA (i.e., the same SPA with different parameter values) result in different interpretations when applied to the same data. An example of the first kind of fault would be the situation where SPA # 1, a power level detector, indicates a power level consistent with the background noise while SPA # 2, Ozturk's distribution identification algorithm, indicates a non-Gaussian distortion. This is a fault because the background noise is assumed to be Gaussian. An example of the second kind of fault would be the situation where use of Ozturk's algorithm based on 100 and 200 samples from the same patch result in a different interpretation.

4.2.2 - <u>Diagnosis and Reprocessing</u>

When the signal being monitored does not satisfy the requirements of the SPA instance, the output of the SPA is distorted resulting in a discrepancy. Once a discrepancy has been detected, a diagnosis procedure is used to identify the distortion that may have led to the discrepancy. Knowing the distortion, then either the appropriate parameters of the same SPA can be adjusted or a different SPA chosen to reprocess the data. In a sense, the diagnosis procedure maps symptoms (discrepancies) to hypothesized underlying causes (distortions). For example, assume the sample mean of a patch is repeatedly being evaluated by processing 100 samples at a time. Although, the first eight trials result in values close to zero, the nineth trial produces a large negative value for the mean. This represents a conflict of the first kind. The diagnosis procedure may surmise that the conflict may be due to the presence of one or more outliers. Consequently, the reprocessing procedure concludes that the data from the ninth trial should be reprocessed using a median detector.

39

4.2.3 - <u>Interpretation Process</u>

The interpretation process is a search through a space of sets of interpretation models for modeling signals. When a possible combinatorial explosion in interpretation models does not exist, the interpretation process may be viewed as just a straight forward classification process. Otherwise, the search must be carried out as a constructive problem solving process. The IPUS architecture employs the constructive problem solving approach which reduces to the classification approach in the absence of a combinatorial explosion.

Constructive problem solving techniques must be used when the set of possible solutions is too large to be enumerated. For example, although the set of PDF types is finite in the radar problem, there are an infinity of different PDF's possible because of the infinity of values that can be assumed by the scale, location, and shape parameter. Consequently, constructive problem solving is needed to approximate the underlying probability distribution of the data.

4.3 - <u>Use of IPUS in A'SCAPE</u>

The IPUS architecture is suitable when a single SPA-instance cannot correctly process all the input signals that can potentially arise in a signal understanding application. In the scene understanding problem, the variety of average magnitude levels in patches and probability distributions underlying the data along with the different tasks to be carried out in A'SCAPE during the monitoring the environment (mapping, statistical processing, and pixel indexing) necessitates more than one SPA-instance. Hence, IPUS is suitable for the scene understanding problem. Next, use of IPUS to control the mapping and statistical procedures is discussed.

4.4 - <u>IPUS and the Mapping procedure</u>

Observations are made next on the different control parameters to set the stage for the design of the different rules to enable IPUS to control the mapping procedure.

4.4.1 - <u>Observations on Setting the Control Parameters</u>

In this section, different effects of the control parameters are discussed. Note first that the intervals for the allowable values of the control parameters are given in Chapter 2.0 and are equal to,

$$0\% \leq LPQP \leq 100\%$$
$$5 \leq NCQ \leq 8 \qquad\qquad (4.1)$$
$$1 \leq NCC \leq 4$$

Recall that LPQP represents the fraction of LP pixels in the quantized volume. It is used to determine the threshold q for which all pixels with data amplitudes below q are identified as LP and all pixels with data amplitudes above q are identified as RPs in the quantized volume.

Also, NCQ is the minimum number of neighboring pixels in the quantized volume required to be identified as RPs pixels for a test pixel to be declared as a RPs pixel in the first-corrected volume. Finally, NCC is the minimum number of neighboring pixels in the first-corrected volume required to be identified as RPs pixels for a test pixel to be declared as a RPs pixel in the second-corrected volume.

LPCQP and LPCCP are computed parameters which represent the LP percentages in the first and second-corrected volumes, CQV and CCV, respectively.

Define LPQPt to be the true value for the fraction of LP pixels in the generated scene.

As explained in Chapter 2 the mapping processor begins by setting a threshold that results in a specified fraction of LP pixels equal to LPQP. The mapping processor iterates until it is satisfied that the latest scene is consistent with the last specified value of LPQP. When the iteration process ends, it is assumed that

$$LPQP \cong LPQPt \qquad\qquad (4.2)$$

4.4.1.a - <u>Observations on the Setting of LPQP</u>

1. **Setting LPQP much smaller than LPQPt:** Many pixels have data amplitudes larger than the threshold resulting in a large number of LP pixels being declared as RPs pixels in the quantized volume.

2. **Setting LPQP much larger than LPQPt:** RPs patch pixels may be misclassified due to the fact that some RPs patches have data amplitude values below the threshold. This results in many RPs pixels being identified as LP pixels in the quantized volume.

   <u>**Conclusion**</u>: The threshold is always set very low at the beginning so that LP information is gained as the process iterates. Because (1) the objective of the mapping procedure is to separate between LP and RPs patches, (2) the average power of the LP is the lowest among all regions, and (3) the threshold is set adaptively by the assessment stage, the threshold, controlled by the assessment stage, is raised adaptively until $LPQP \cong LPQPt$.

4.4.1.b - <u>Observations on the Setting of NCQ</u>

   Recall that NCQ controls which test pixels in the first-corrected volume are to be declared as RPs. NCQ is said to be large when its value approaches 8 and small when its value approaches 5. The following observations relative to NCQ take into consideration that the initial setting of LPQP is low and then is increased until LPQP approximates the true value LPQPt. Depending on the setting of LPQP with respect to LPQPt, four cases exist:

1. **LPQP much smaller than LPQPt:**

   a) **Setting NCQ small:** In this case, because many RPs declared pixels exist in the quantized volume due to the low threshold, small NCQ results in the building of a multitude of RPs patches which are likely to be so close that they form a single big RPs patch in the first-corrected volume.

   b) **Setting NCQ high:** Here, even though many RPs declared pixels exist in the quantized volume due to the low threshold, high NCQ results in the building of fewer RPs patches than when NCQ is small. This is due to the fact that

there must be at least NCQ RPs pixels neighboring the test pixel in the quantized volume, where NCQ is large, in order for the test pixel to be declared as a RPs pixel in the first-corrected volume. In this case, corrections are made and some of the pixels previously declared as RPs pixels in the quantized volume are now declared as LP pixels in the first-corrected volume.

2. **LPQP close to LPQPt:** When LPQP is close to its true value, the threshold is high enough to separate between the LP region and RPs patches. With either small or large values for NCQ, the RPs regions are well approximated. In this case, the choice of NCQ affects the classification of the inner pixels of the RPs regions. This results because, even though the data amplitudes of RPs pixels are higher than those of the LP pixels, in general, some RPs pixels with data amplitudes lower than those of the highest LP data values exist and may be lower than the threshold.

   a) **Setting NCQ small:** All test pixels in the quantized volume that have at least NCQ neighboring pixels are declared as RPs pixels in the first-corrected volume. Small NCQ helps to correctly classify the inner RPs pixels. However, note that small NCQ also results in misclassifying LP pixels that are surrounded by at least NCQ declared RPs pixels.

   b) **Setting NCQ high:** Every test pixel must have a large number of neighboring RPs declared pixels in the quantized volume for it to be declared as a RPs pixel in the first-corrected volume. This causes the procedure to misclassify some of the inner RPs pixels when too many of the neighboring pixels have their data amplitudes falling below the threshold. In this case, the identified RPs regions are not homogeneous and contain LP declared "holes".

   <u>Conclusion:</u> The value of NCQ should be chosen as large as possible at the beginning of the iterative process when the threshold is set very low to correctly reclassify the maximum number of LP pixels misidentified at the thresholding/quantization stage. When the threshold reaches a level where it is close to its convergence value, NCQ should then be chosen small to avoid non-homogeneous RPs regions.

4.4.1.c - <u>Observations on the Setting of NCC</u>

Because NCQ truncates the boundaries of the RPs regions, NCC is used to augment the edges of the RPs declared regions. NCC is said to be large when its value approaches 4 and small when its value approaches 1. In the following discussion it is assumed that the conclusions previously reached on the settings of LPQP and NCQ are taken into consideration so that LPQP is initially set low to be increased until it approaches its true value LPQPt, while NCQ is initially set to a large value, to be decreased as LPQP approaches its true value. Four cases are then identified:

1. **LPQP much smaller than LPQPt and NCQ large:** Because NCQ is set large, many RPs edge pixels are misclassified and associated with the LP region.

   a) **Setting NCC small:** When NCC is set small, many of the edge pixels are correctly reclassified from LP pixels to RPs pixels in the second-corrected volume.

   b) **Setting NCC large:** In this case, only a few misclassified RPs edge pixels are correctly reclassified in the second-corrected volume.

2. **LPQP close to LPQPt and NCQ small:** Because NCQ is small, only a few RPs edge pixels are associated with the LP.

   a) **Setting NCC small:** Small NCC causes not only RPs edge pixels to be recovered but also LP pixels to be misclassified in the second-corrected volume.

   b) **Setting NCC large:** In this case, most of the RPs edge pixels are correctly classified in the second-corrected volume and only few LP pixels are misclassified as RPs pixels.

<u>Conclusion:</u> NCC results in the recovery of RPs edge pixels and the misclassification of some LP pixels close to the RPs edge pixels. In order to maximize recovery of the RPs edge pixels and minimize the misclassification of LP pixels, NCC should be set small when NCQ is set large in order to recover many of the RPs edge pixels that were lost in the first-correction. On the other hand, NCC should be set large

44

when NCQ is set small because, in this case, only a few RPs edge pixels need to be recovered.

4.4.2 - <u>Resolution of Discrepancies</u>

In this section, rules are developed to enable the resolution of discrepancies. The assessment stage of the mapping procedure consists of comparing at each step of the iteration the value for LPCCP with the corresponding LPQP. When LPCCP is not sufficiently close to LPQP, the assessment stage is said to fail. This initiates the discrepancy detection stage. Diagnosis identifies the distortion that may have caused the discrepancy and adjusts one or more of the mapping control parameters for reprocessing of the data.

The strategy behind the iterative process of the mapping procedure employs two stages. In the first stage, referred to as the threshold approximation stage, LPQP is varied iteratively by the mapping processor until, as explained later, it is expected that LPQP is within 10% of its true value LPQPt. The second stage, referred to as the threshold fine-tuning stage, consists of iteratively varying LPQP until it converges to within 1% of the last computed value for LPCCP. The two stages are now discussed in detail.

4.4.2.a - <u>Discrepancies in the Threshold approximation stage</u>

During this stage, two sets of SPA instances are used on the same data of the surveillance volume. For both sets LPQP and NCC are the same whereas NCQ is equal to 7 for one set and 8 for the other.

In order to understand how the rules are set in this stage, it is important to subdivide the problem into two cases. Case I describes the situation where the LP and RPs patches have histograms with large area of overlapping whereas case II describes the situation where the LP and RPs patches have histograms with small area of overlapping to non-overlapping.

<u>Case I</u> - **LP and RPs whose histograms have large area of overlap**

Recall that NCQ is used to recognize the RPs patches in the surveillance volume. First consider the situation where LPQP approximately equals LPQPt. Here the threshold

is such that it is possible to do a good job of separating between the LP region and RPs patches. Steps are then taken to correct misclassified LP and RPs data. Note that misclassifications are due to large LP data exceeding the threshold and small RPs data falling below the threshold. At this point, setting NCQ to 7 and 8, respectively, results in very close values for LPCQP and LPCCP due to the facts that (1) the two masks are very similar (NCQ=8 requires that 8 neighboring pixels be declared RPs in the quantized volume for a test pixel in the first-corrected volume to be declared RPs whereas NCQ=7 requires that 7 neighboring pixels be declared RPs in the quantized volume for a test pixel in the first-corrected volume to be declared RPs), (2) only a few pixels are misclassified in the quantized volume.

Now consider that LPQP is significantly smaller than LPQPt. In this case many LP pixels are misclassified after quantization. Even though masks with NCQ equal to 7 and 8 are similar, they result in LPCQP and LPCCP being considerably different due to the fact that the large number of misclassified LP pixels are so many that they tend to group together. Consequently, changing NCQ from 8 to 7 simply results in additional LP pixels grouping together to form additional RPs regions and more edges. Because of this,

$$LPCQP|_{NCQ=7} < LPCQP|_{NCQ=8}$$
$$\text{and} \qquad\qquad\qquad (4.3)$$
$$LPCCP|_{NCQ=7} < LPCCP|_{NCQ=8} \; .$$

Because the second-corrected volume represents the scene where RPs patches and their edges are assumed to be properly recovered had the threshold LPQP been chosen properly, LPCCP tries to converge to LPQPt. Thus, it is logical to begin each iteration by assigning to LPQP the latest computed value of $LPCCP|_{NCQ=8}$. The very first value assigned to LPQP is simply a guess. This value should be such that the threshold is low. In all of our examples, the first value of LPQP is chosen equal to 10%.

Using different scenes with different values for LPQPt, it has been determined near convergence that whenever the difference between $LPCCP|_{NCQ=7}$ and $LPCCP|_{NCQ=8}$ is within 10%, then LPQP is likely to be within 10% of its true value. This is confirmed in

46

Figure 4.1 where plots of the quantities $\text{LPCCP}|_{\text{NCQ=8}}$ - $\text{LPCCP}|_{\text{NCQ=7}}$ versus LPQP-LPQPt are shown for different values of LPQPt. Note that

- **i** - for different values of LPQPt, all plots are such that near convergence the difference $\text{LPCCP}|_{\text{NCQ}}=8$ - $\text{LPCCP}|_{\text{NCQ}}=7$ approaches zero when LPQP-LPQPt also approaches zero,

- **ii** - for different values of LPQPt, when LPQP-LPQPt > -10%, the difference $\text{LPCCP}|_{\text{NCQ}}=8$ - $\text{LPCCP}|_{\text{NCQ=7}}$ < 10%.

As a result of the above, the threshold approximation stage iterates until it is satisfied that

$$\text{LPCCP}|_{\text{NCQ=8}} - \text{LPCCP}|_{\text{NCQ=7}} < 10\%. \quad (4.4)$$

<u>Case II</u> - LP and RPs whose histograms either have a small area of overlap or do not overlap at all

Equation 4.4 is mostly valid when the histograms of the different patches overlap noticeably. When the histograms do not overlap or have a very small region of overlap, it is noticed that the difference in Equation 4.4 is even less than 10% for all settings of LPQP, when LPQP is significantly smaller or larger than LPQPt many LP pixels are misclassified after quantization. Even though masks with NCQ equal to 7 and 8 are similar, they result in LPCQP and LPCCP being considerably different due to the fact that the large number of misclassified LP pixels are so many that they tend to group together. Consequently; changing NCQ from 8 to 7 simply results in additional LP pixels grouping together to form additional RPs regions and more edges causing the difference of $\text{LPCCP}|_{\text{NCQ=8}}$ - $\text{LPCCP}|_{\text{NCQ=7}}$ to be large. On the other hand, when LPQP is very close to LPQPt the threshold is set so that it is possible to separate between the LP and RPs, the setting of NCQ to 7 and 8, respectively, results in the difference of $\text{LPCCP}|_{\text{NCQ=8}}$-$\text{LPCCP}|_{\text{NCQ=7}}$ being very small due to the fact that only a few pixels are misclassified in the quantized volume.

Using the observations above, it is concluded that in order to find the threshold to separate between LP and RPs when their histograms do not overlap it is sufficient to look for the minimum in the difference $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ as LPQP is varied.

In summary, a guess for the initial value of LPQP is followed by the execution of the mapping procedure using two different SPA instances. The outputs of the two SPA instances are compared by means of the computed values of LPCCP. If $LPCCP|_{NCQ=8}$-$LPCCP|_{NCQ=7}$ is more than 10%, a discrepancy is detected and it is concluded that the value of LPQP differs from its true value by more than 10%. LPQP is then increased to the latest computed value of $LPCQP|_{NCQ=8}$. LPQP is varied from one iteration to the next while NCC is kept equal to 1. This choice for NCC agrees with the observations made
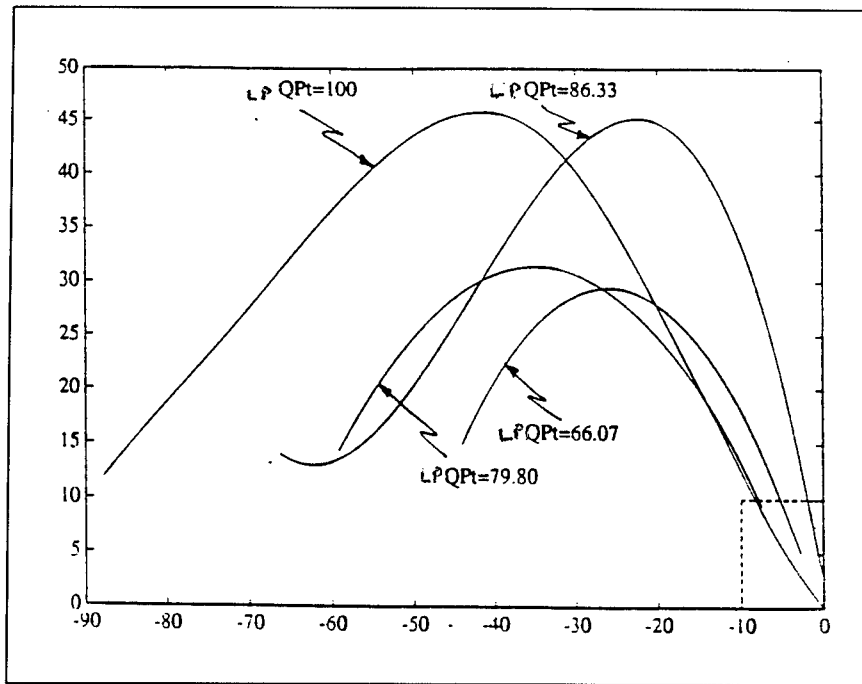

Figure 4.1 - <u>Plot of $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ Versus LPQP-LPQPt for Different Values of LPQPt</u>

previously where it was concluded that NCC should be set small when NCQ is set large. In this case NCQ has a large value equal to either 7 or 8.

The discrepancies that may arise in the threshold approximation stage are due to the fact that two instances of the same SPA result in different interpretations when applied to the same data. As defined in Section 4.2.1, such a discrepancy is typified as a fault.

Four fault-type discrepancies are readily identified in the threshold approximation stage. These are as follows:

1. **LPCCP$|_{NCQ=8}$ - LPCCP$|_{NCQ=7}$ > 10%:** As discussed above, the goal of the threshold approximation stage is to obtain a threshold LPQP that is within 10% of its true value. As shown in Figure 4.1, this is likely only when LPCCP$|_{NCQ=8}$ - LPCCP$|_{NCQ=7}$ < 10%. When the difference between the computed thresholds LPCCP$|_{NCQ=8}$ - LPCCP$|_{NCQ=7}$ is more than 10%, a fault type of discrepancy is detected during the assessment stage. The diagnosis process identifies the fact that LPQP-LPQPt < -10% as the source for the distortion causing the discrepancy. The remedy, in this case, is to increase the value of LPQP during the reprocessing stage to the latest computed value for LPCCP$|_{NCQ=8}$.

2. **initial LPQP set too low:** In some cases, when the initial guess for LPQP is too small, the number of LP pixels with data exceeding the threshold is so large that when corrections are made, the second-corrected volume results in many RPs declared patches or, in the worst case, a single big RPs patch. This results in the values of either LPCCP$|_{NCQ=8}$ or LPCCP$|_{NCQ=7}$ being even smaller than LPQP. In this case, it is possible to obtain a value for the difference LPCCP$|_{NCQ=8}$ - LPCCP$|_{NCQ=7}$ that is smaller than 10%. The IPUS control must be suspicious of such a case and declare LPQP-set-too-low as the source for the distortion causing the discrepancy. The remedy, in this case, is to increase the initial value of LPQP during the reprocessing stage. In our examples, we choose to increase LPQP by 10% every time an initial-LPQP-set-too-low fault is obtained.

3. **$LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ < 10% even when LPQP is increased:** The IPUS control must be suspicious of such a case and declare the fact that the histograms of the LP and RPs do not overlap as the source for the distortion causing the discrepancy. The remedy, in this case, increase LPQP by 10% steps until a minimum value for $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is obtained.

4. It is noted that during the threshold approximation stage it is possible that the **inequality in Equation 4.4 will be satisfied with $PLCCP|_{NCQ=7\ or\ 8}$ =100%.** This means that the subpatch with the smallest average power occupies 100% of the RPs patch area. Consequently, the RPs patch is homogeneous. When the inequality in Equation 4.4 is met with $PLCCP|_{NCQ=7}$ =100%, there is no need for the threshold fine-tuning stage. This is because the initial value of PLQP in the threshold fine-tuning stage would be equal to 100% and any more processing would also result in PLCCP=100% regardless of the values chosen for NCQ and NCC.

Table 4.1, summarizes the discrepancies that may occur during the threshold approximation stage.

Discrepancy (Fault Type)	Diagnosis	Reprocessing
$LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is more than 10%	PLQP-PLQPt>-10%	Assign to PLQP the latest value of $PLQP|_{NCQ=8}$
$LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is less than 10% in the early stages of iteration	Initial value of PLQP is too low	Increase PLQP by 10% from its initial value
$LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is less than 10% in all iterations	LP and RPs histograms do not overlap	Search for the PLQP that results in a minimum value $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$
Inequality in Eq. (VII.4-1) will be satisfied with $PLCCP|_{NCQ=7\ or\ 8}$ =100%	The scene is homogeneous	Declare that no LP exist. No threshold fine-tuning is needed

Table 4.1 - <u>Discrepancies in the Threshold Approximation Stage</u>

4.4.2.b - <u>Threshold fine-tuning stage</u>

At the end of the threshold approximation stage LPQP is likely to be within 10% of its true value LPQPt. During the threshold fine-tuning stage, LPQP is varied until LPCCP is within 1% of the corresponding value of LPQP.

During the threshold fine-tuning stage, NCQ is lowered to avoid holes in the RPs patches caused by misclassified RPs pixels whose data values are lower than the threshold. For the same reason LPQP is assigned the latest value of $LPCCP|_{NCQ=7}$ rather than the latest value of $LPCCP|_{NCQ=8}$. In addition, the value of NCC is raised to avoid misclassification of LP pixels close to the RPs edges. These choices for NCQ and NCC agree with observations mentioned in Section 4.4.1.

The following observations on LPQP, NCQ, and NCC are necessary to understand how these parameters should be automatically set in order for LPCCP to converge to within 1% of LPQP.

1 - When LPQP is increased while both NCQ and NCC are kept constant, the number of LP pixels in the quantized volume is increased and, therefore, both LPCQP and LPCCP are likely to increase.

2 - When NCQ is increased while both LPQP and NCC are kept constant, the requirement on a test pixel to be declared as a RPs pixel in the first-corrected volume becomes more stringent and, therefore, the number of RPs pixels in both corrected volumes are likely to decrease. This tends to increase the number of LP pixels causing both LPCQP and LPCCP to increase.

3 - When NCC is increased while both LPQP and NCQ are kept constant, the requirement on a test pixel to be declared as a RPs pixel in the second-corrected volume becomes more stringent and, therefore, the number of RPs pixels in this volume is decreased. Thus, the number of LP pixels in the second-corrected volume increases and, consequently, LPCCP increases.

Using the above observations, the following strategy is used by the assessment stage to control the threshold fine-tuning stage,

1 - Because LPQP is within 10% of its true value at the beginning of the threshold fine-tuning stage, the threshold is likely to be relatively high. Thus, NCQ should be set to its smallest value of 5 while, as needed, NCC should be incremented iteratively from its minimum value of 1 up to its maximum value of 4.

2 - When the inequality in Equation 4.4 is not satisfied, LPQP should be increased in small steps. Otherwise, the iterative process diverges when the same rule from the threshold approximation stage is used. The approach taken in this work during the threshold fine-tuning stage consists of assigning a value to LPQP that is half way between its latest value and the latest value of LPCCP, i.e.

$$LPQP = \frac{LPQP_{latest} + LPCCP_{latest}}{2} \tag{4.5}$$

3 - The condition set forward for ending the threshold fine-tuning stage is given by

$$|LPQP - LPCCP| < 1\% \tag{4.6}$$

Two cases are possible when the inequality in Equation 4.6 is not satisfied: either LPQP < LPCCP or LPQP > LPCCP.

4 - When the inequality in Equation 4.6 is not satisfied and LPQP < LPCCP, the control parameters should be varied by the diagnosis procedure such that LPCCP is decreased. In this case, NCQ should be made smaller. If none of the allowable values for NCQ result in the inequality in Equation 4.6 being satisfied, then LPQP is varied according to Equation 4.5.

5 - When the inequality in Equation 4.6 is not satisfied and LPQP > LPCCP, the control parameters should be varied by the diagnosis procedure such that LPCCP is increased. In this case, NCC should be made larger. If none of the allowable values for NCC result in the inequality in Equation 4.6 being satisfied, then LPQP is varied according to Equation 4.5.

52

6 - If the threshold fine-tuning stage results in PLCCP=100% at any iteration, the threshold fine-tuning stage should end because, as in observation 2-, any more processing will end with PLCCP=100% regardless of the values chosen for NCQ and NCC. This, in turn, will make PLQP equal to 100%.

Note that during the threshold approximation stage note that only LPQP was varied whereas, in the threshold fine-tuning stage, any of the parameters LPQP, NCQ, and NCC may be varied.

In summary, the threshold fine-tuning stage begins by assigning to LPQP the latest value of $LPCCP|_{NCQ=7}$. Once quantization, first-correction, and second-correction stages are completed with pre-selected values for NCQ and NCC, the assessment stage diagnoses the results according to the strategy discussed above, and, depending on the outcome, decides either that reprocessing is necessary with adjusted values for any of the LPQP, NCQ, and NCC parameters, or the threshold fine-tuning stage is completed.

At the end of each iteration of the threshold fine-tuning stage it is expected that the computed value of LPCCP will be within 1% of LPQP. When the inequality in Equation 4.6 is not satisfied, a conflict type of discrepancy is detected based on the inconsistency in the expectation that LPCCP will be within 1% of LPQP. Table 4.2 summarizes the discrepancies that may occur during the threshold fine-tuning stage.

Discrepancy (Conflict Type)	Diagnosis	Reprocessing
|PLQP-PLCCP|>1% and PLQP<PLCCP	Either NCQ or PLQP are not well adjusted	Decrease NCQ, otherwise update LPQP
|PLQP-PLCCP|>1% and PLQP>PLCCP	Either NCC or PLQP are not well adjusted	Increase NCC, otherwise update PLQP

Table 4.2 - <u>Discrepancies in the Threshold Fine-Tuning Stage</u>

It has been determined through examples that the initial setting of NCQ=5 is adequate for the threshold fine-tuning stage to converge (i.e., in the different treated examples, it was never necessary to decrease NCQ from the value of 5).

4.5 - IPUS and the Statistical Procedure

Recall that the objective of the statistical procedure is to separate between contiguous non-homogeneous patches based on their data distributions. A four step strategy for the statistical procedure stage was presented in Chapter 3. The steps are:

1. Using the goodness of fit test of the Ozturk algorithm, a Gaussianity check is performed on every tile to ensure whether the data in the tile are Gaussian or not. This results on every patch having its tiles labeled as either Gaussian or non-Gaussian.

2. Existing outliers are located in those tiles declared as non-Gaussian.

3. For every non-Gaussian declared tile, pixels with outliers are excised from the tile and replaced with the closest pixels to the tile whose data are not outliers. Next, the Gaussianity check is performed once again as in step (1). At the end of this step, each patch will have tiles declared as Gaussian or non-Gaussian and pixels with outliers.

4. Using the Ozturk algorithm, the (U,V) coordinates of the locus end point is obtained for every tile declared as non-Gaussian in step (3). Next, a check is made to ensure whether or not the data of the set of tiles which constitutes a sub-patch can fit within a confidence ellipse. This is done by first declaring a subpatch every set of contiguous non-Gaussian tiles. Then, computing the average (Uav,Vav) coordinates of all test pixels of the same subpatch and getting its best approximating PDF and the corresponding confidence ellipse. Finally, a check is made whether all (U,V) coordinates of the test pixels are within the confidence ellipse of the average coordinates (Uav,Vav). If not, the tiles are regrouped so that all (U,V) coordinates for each group of tiles can fit within the same ellipse. Each group then forms a subpatch.

IPUS is primarily needed in the fourth step of the statistical procedure where a grouping of contiguous tiles whose (U,V) pairs can fit under a single ellipse is to be performed to form subpatches. This step necessitates rules to enable the diagnosis process

to acknowledge that a discrepancy exists that is caused by the distortion due to the contiguous non-Gaussian tiles not being all homogeneous. This distortion is displayed when a set of contiguous non-Gaussian tiles cannot have all of their (U,V) pairs fit under the same ellipse.

Then, reprocessing takes place to find sets of homogeneous non-Gaussian tiles whose (U,V) pairs can fit under the same ellipse and declare each set as a subpatch. This is done under the constraint that a minimum number of ellipses should be used. Note that the discrepancy in this case is of conflict type because the expectation that a set of contiguous non-Gaussian tiles should be homogeneous is not met. Table 4.3 summarizes the discrepancy in the statistical procedure.

Discrepancy (Conflict Type)	Diagnosis	Reprocessing
A set of contiguous non-Gaussian tiles cannot have all of their (U,V) pairs fit under the same ellipse	Not all of the contiguous non-Gaussian tiles are homogeneous	Find sets of homogeneous tiles whose (U,V) pairs can fit under the same ellipse and declare each set as a subpatch.

Table 4.3 - Discrepancy in the Statistical Procedure

4.6 - Conclusion

The expert system IPUS is used to guarantee the convergence of both the mapping and statistical procedures. Through its rules for reprocessing IPUS controls all feed-forward and feed-back connections between the different blocks of A'SCAPE.

5

Application of A'SCAPE to real IR data

5.1 - Introduction

In this chapter, real data of an IR image are processed through A'SCAPE. Section 5.2, discusses how the data is collected. Sections 5.3 through 5.6 present the results of processing the IR scene through the different stages of A'SCAPE. Finally, section 5.7 consists of a conclusion for real data processing.

5.2 - Data Collection

The data processed in this work consist of real airborne data of infrared (IR) type collected over lake Michigan. As shown in Figure 5.1[11] (reproduced from the reference mentioned in the footnote), the airborne data is collected through an m-7 line scanner which consists of an optical telescope with its narrow field of view (ground resolution element) redirected by a rotating flat mirror causing the system to scan in a plane perpendicular to the longitudinal axis of the aircraft. A radiation detector in the focal plane of the telescope converts the focused beam of radiation to an electrical signal. The optical system's field of view first scans laterally across the aircraft. Then, before making the next ground scan, it scans radiation references which are internal to the scanner. By the time the next scan begins, the aircraft has moved forward; thus subsequent line scans build upon one another to produce a continuous strip image of the terrain beneath the aircraft. Figure 5.1 shows the scanner looking directly downward.

5.3 - Preprocessing Stage

Because the univariate approximation chart of the Ozturk algorithm is used to approximate the PDF of different regions the collected data is therefore required to result

---

[11] A. J. La Rocca and D. J. Witte, "Handbook of the Statistics of Various Terrain and Water (ICE) Backgrounds From Selected U.S. Locations," Final Report. Environmental Research Institute of Michigan. Contract No. N60530-79-R-0036. January 1980.
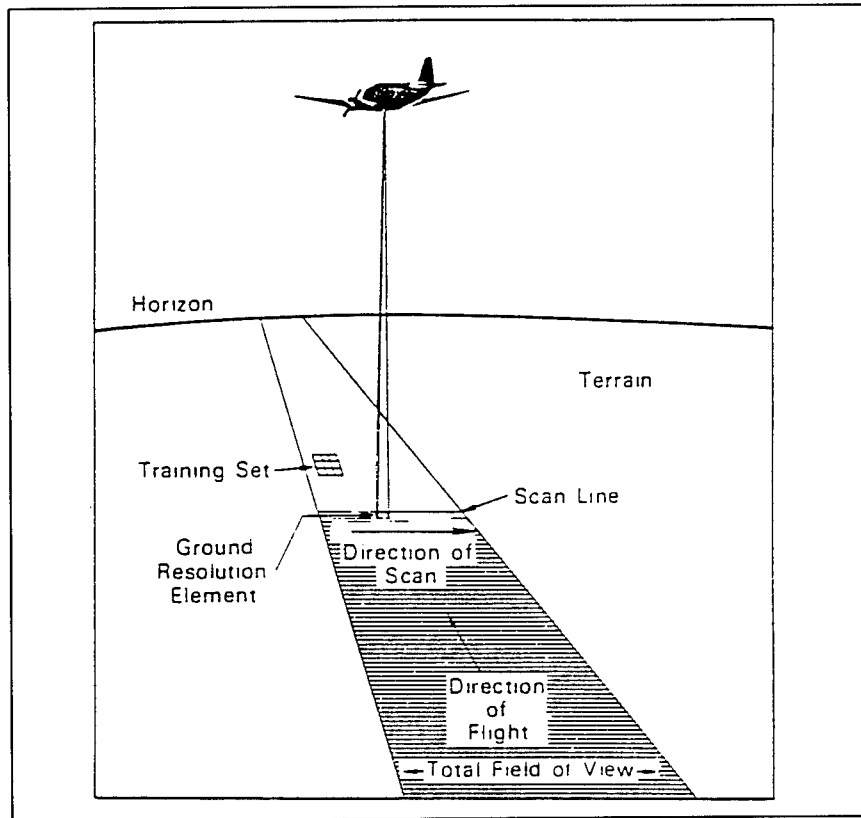
Figure 5.1 - <u>Airborne Scanner Operation</u>

in independent samples. Since there are no tests available to determine the statistical independence of a set of data, the best solution is to ensure that the data are spatially uncorrelated in both width and length. This is done in the preprocessing stage of the A'SCAPE procedure.

The autocovariance function has been calculated for both sets of data in the width direction and the length direction. Examples are shown in Figure 5.2 of the lag centered at the 10th width pixel for all pixels in the width direction for the 10th length pixel and in Figure 5.3 of the lag centered at the 50th pixel for all pixels in the length direction for the 10th width pixel. It has been determined that in order to obtain uncorrelated data, decimation is necessary where only 1 sample out of every 5 samples is retained in the width direction and 1 sample out of every 4 samples is retained in the length direction reducing the scene size from a high resolution 383x646 one to one with a lower 96x130 resolution.
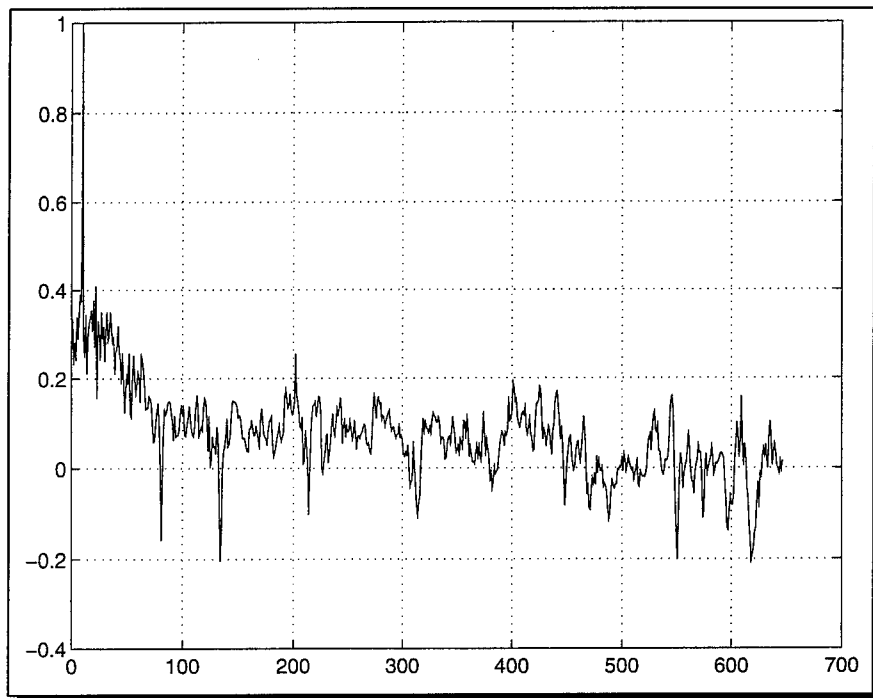
Figure 5.2 - <u>Autocovariance Function in Width for the 10th Length Pixel Centered at the 10th Width Pixel</u>
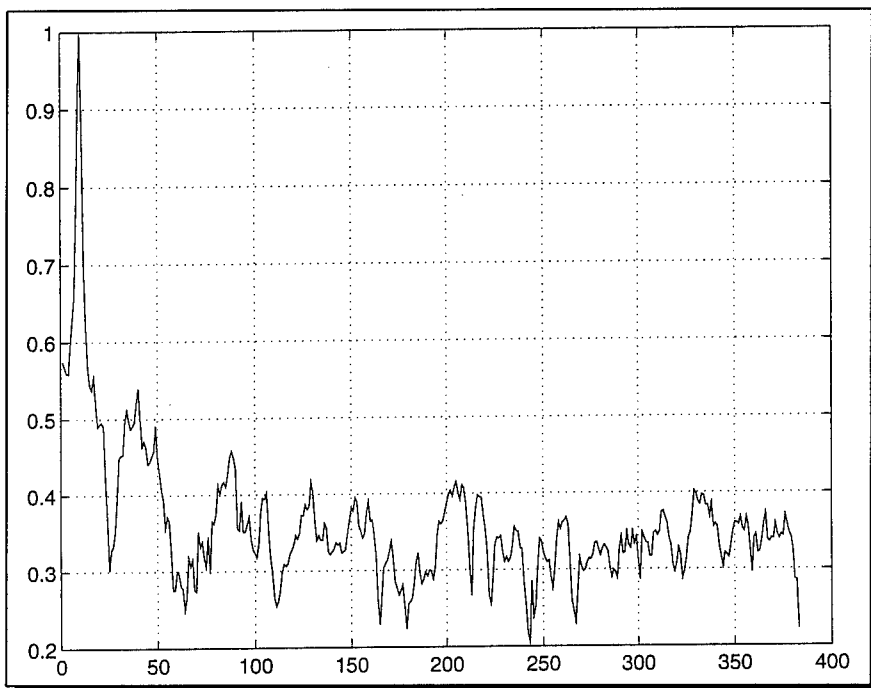
Figure 5.3 - <u>Autocovariance Function in Length for the 50th Width Pixel Centered at the 10th Length Pixel</u>

Note that the decimation is necessary for the statistical procedure part of this work and is not necessary for the mapping procedure. Nevertheless, only the data in the decimated scene are treated in this work for the sake of processing time.

The data processed in this work and shown in Figure 5.4 is collected over lake Michigan and consists of a 2829 ft wide by 1650 ft long scene that contains two major regions: lake and land. The data is 8 bits and the resulting image size when the pre-processing stage is completed is 96x130. Furthermore, the three dimensional magnitude plot of the scene, shown in Figure 5.5, reveals that the data in the lake is regular and similar in magnitudes whereas the data in the land region is irregular and bears a lot of discretes especially near the boundary between the land and the lake. This latter fact is due to the non-homogeneity of the land region which contains a lot of non-homogeneous regions due to the presence of trees, roads, etc.

Once the pre-processing stage is completed, the data is forwarded to the mapping stage for partitioning.
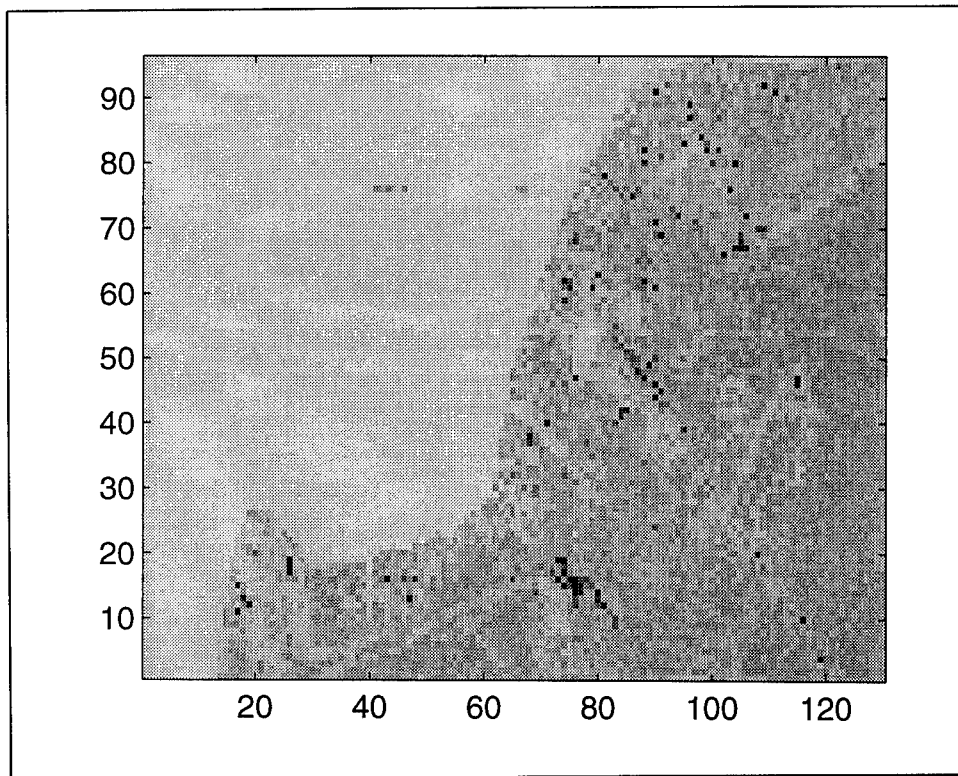
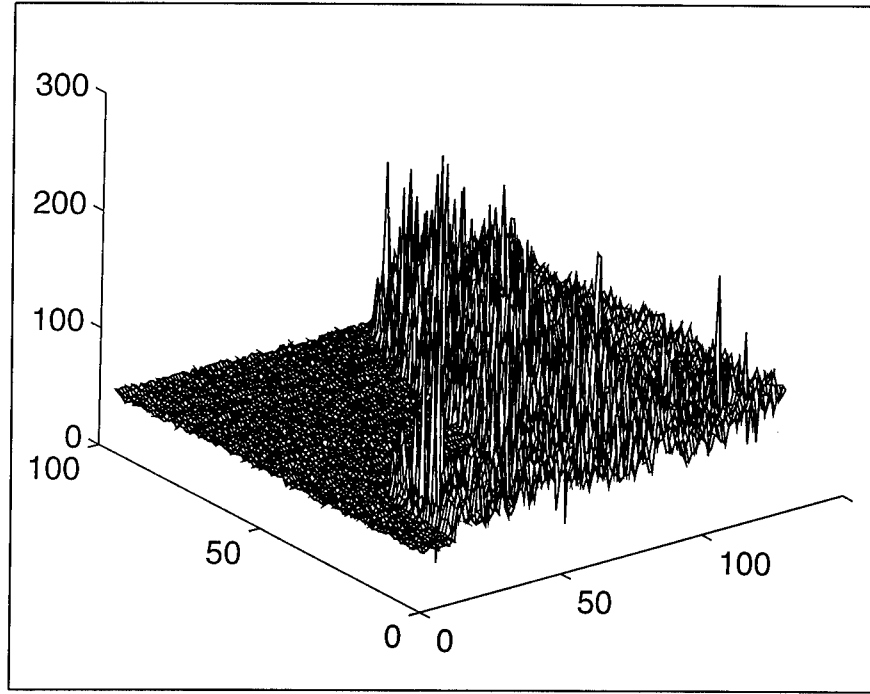**Figure 5.4 - <u>Original Scene (Over Lake Michigan)</u>**

Figure 5.5 - <u>Three Dimensional Magnitude Plot of the Original Scene</u>

5.4 - <u>Mapping Procedure Stage</u>

After the preprocessing block has identified the uncorrelated data in the scene, the mapping procedure is used to separate contiguous non-homogeneous regions with different average magnitudes.

When the iterative procedure starts, the diagnosis procedure detects a discrepancy due to the fact that the difference $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is less than 10% in all iterations. In this case, and as shown in Figure 5.6, LP (Lake) and RPs (Land) histograms have a small area of overlap. As summarized in Table 4.1, the system searches for the PLQP that results in a minimum value $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$.

After seven iterations of the "Identification of LP and RPs" stage the threshold is accepted and the assessment passes. Table 5.1 summarizes the values of the different parameters set in the Thresholding/Quantization, First and Second Correction stages as described in Chapter 4. Note that the minimum value for $LPCCP|_{NCQ=8}$ - $LPCCP|_{NCQ=7}$ is
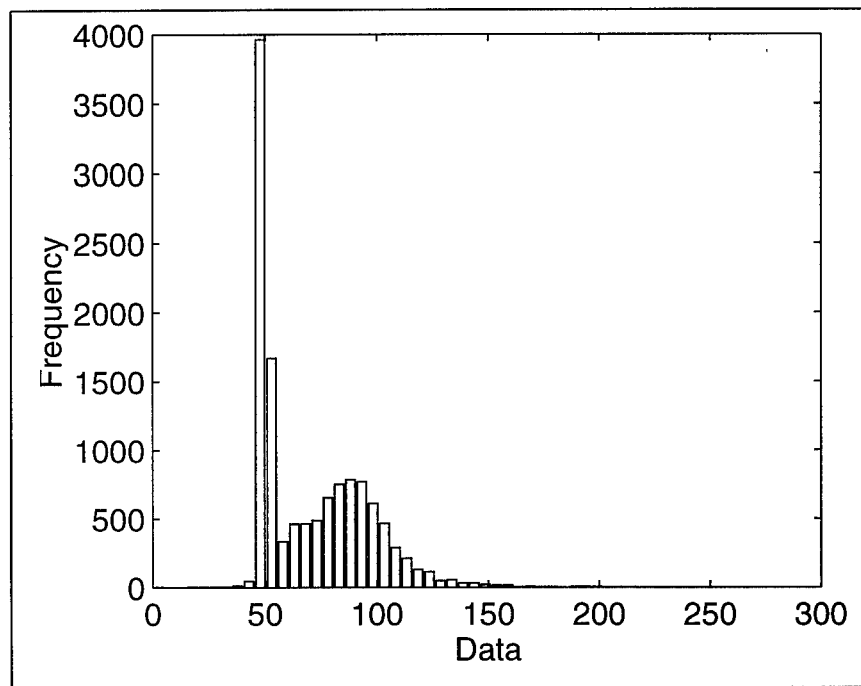
Figure 5.6 - <u>Histogram of the Original Scene</u>

obtained for PLQP=40% (step 4). Note also that steps 1 to 5 are part of the threshold approximation stage whereas steps 6 and 7 are part of the threshold fine-tuning stage.

Figures 5.7 to 5.11 show the plots of the QV (top right), CQV (bottom left), and CCV (bottom right) scenes for steps 1, 2, 3, 4, and 7, respectively. Note how the quantization results more and more in a correct separation between the lake and land as PLQP is increased. The "Identification of LP and RPs" stage is followed by the "Detection of Patch Edges" stage for which the steps of the last CCV (top left), SV (top right), enhanced edges volume (bottom left), and detected edges volume (bottom right), are shown in Figure 5.12 resulting in the segmentation of the scene into three different patches and their respective boundaries. Note that in addition to patches 1 and 2, corresponding respectively to land and lake, the mapping procedure detected a third region, labeled 3. This region can be sighted in the original scene of Figure 5.4 and is not large. In fact Table 5.2 shows that patch 3 contains only 15 pixels as opposed to patches 1

Step	LPQP (%)	NCQ	NCC	LPCCP (%)	$\text{LPCCP}|_{NCQ=8} - \text{LPCCP}|_{NCQ=7}$ (%)
1	10	8	1	24.59	8.72
2	20	8	1	35.82	6.93
3	30	8	1	43.96	4.91
4	40	8	1	46.57	1.67 ✓
5	50	8	1	56.31	5.98
6	45	5	1	45.14	-
7	45	5	2	45.63	-

Table 5.1 - <u>Setting of the Parameters in the Thresholding/Quantization, First and Second Correction stages</u>

	Patch 1	Patch 2	Patch 3
Number of pixels	6680	5337	15
Mean	87.61	49.47	34.4
Variance	420.20	4.13	7.41
Average Power (dB)	39.08	33.89	34.40

Table 5.2 - <u>Mapping Procedure Assessment</u>

First Iteration

Quantized Volume at 10 %

LPQP= 10 %

NCQ = 8
NCC = 1

LPCCP= 24.59 %

First Corrected Volume

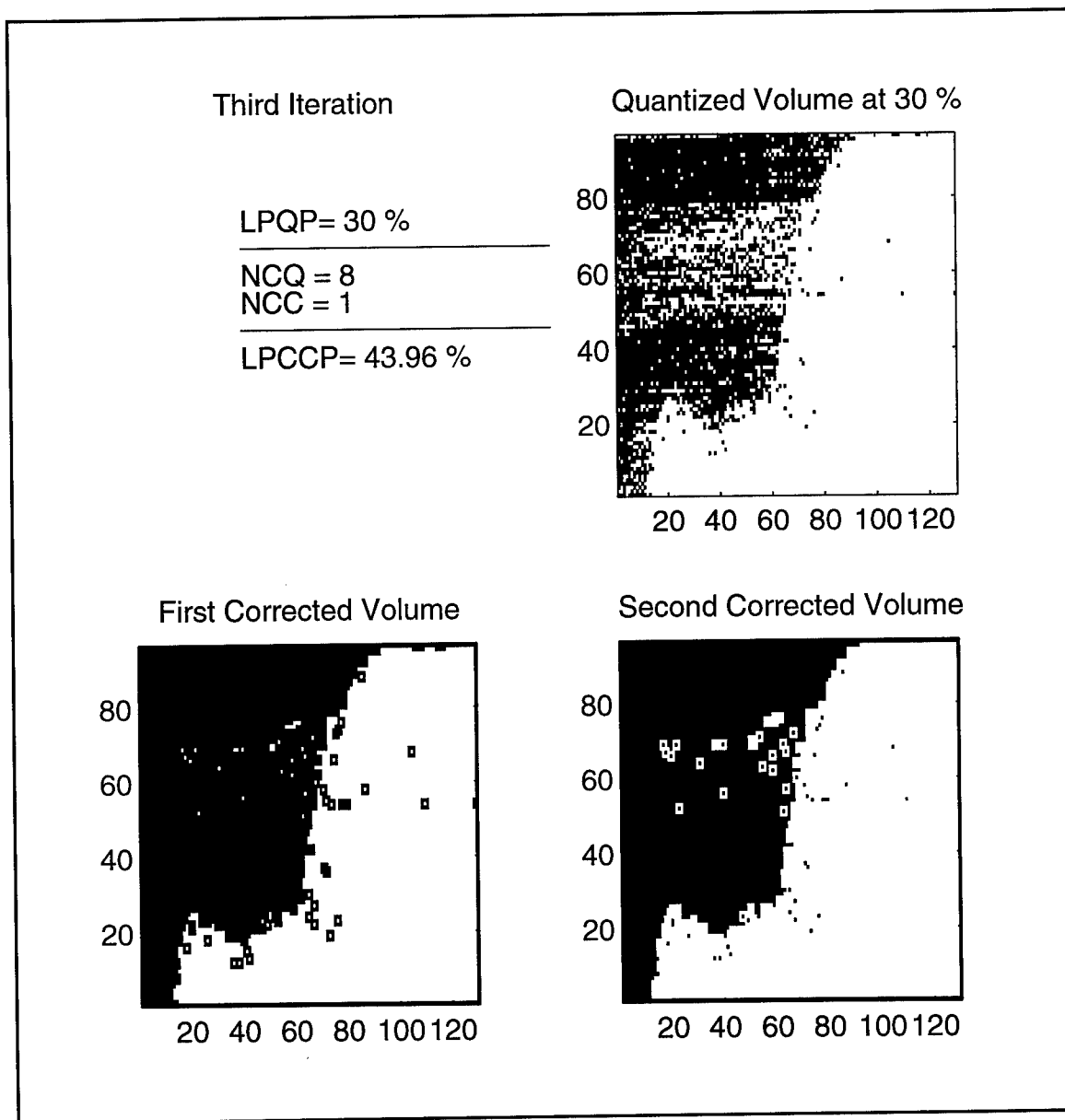Second Corrected Volume
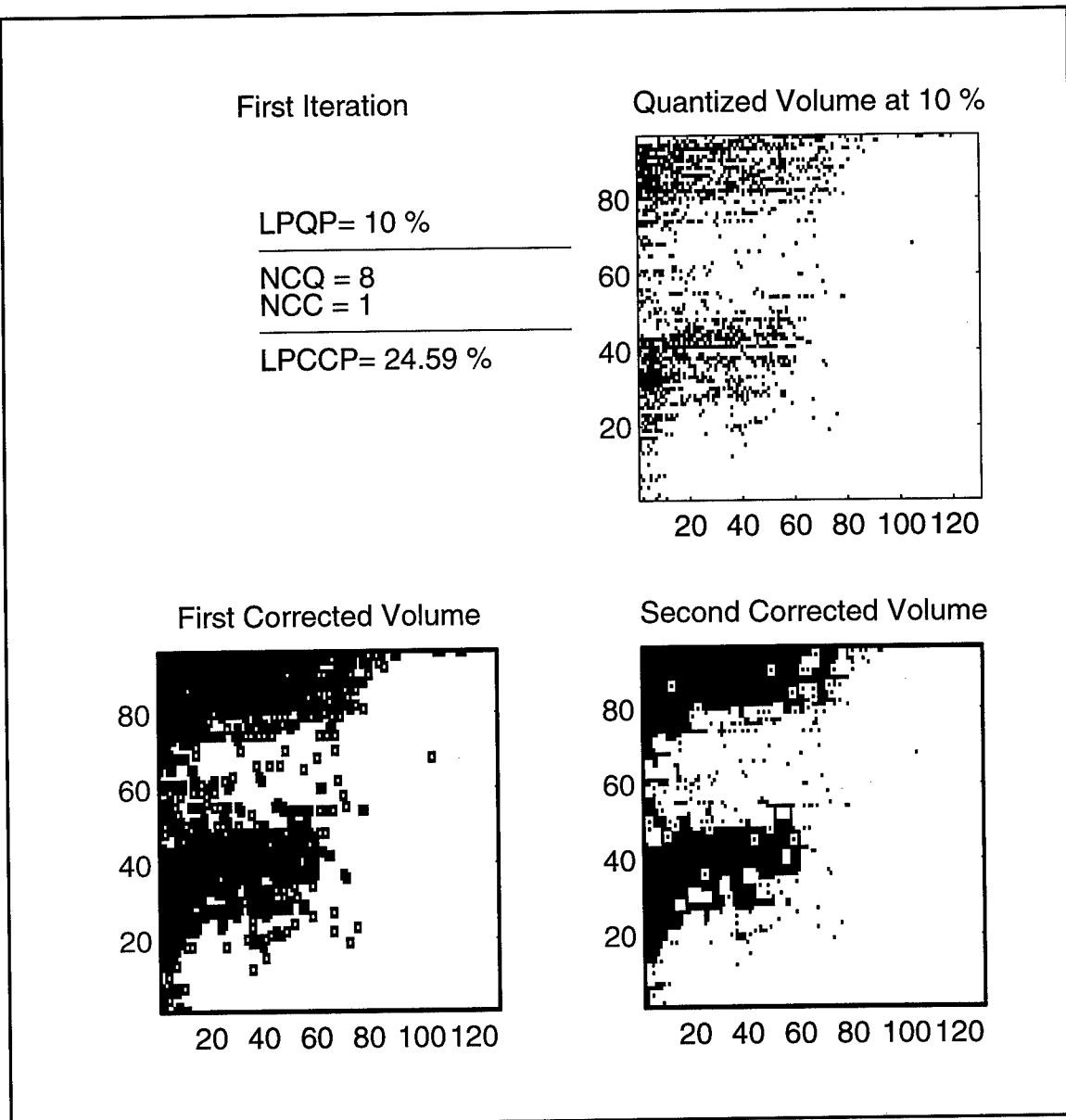
Figure 5.7 - Step 1 of the Identification of LP and RPs
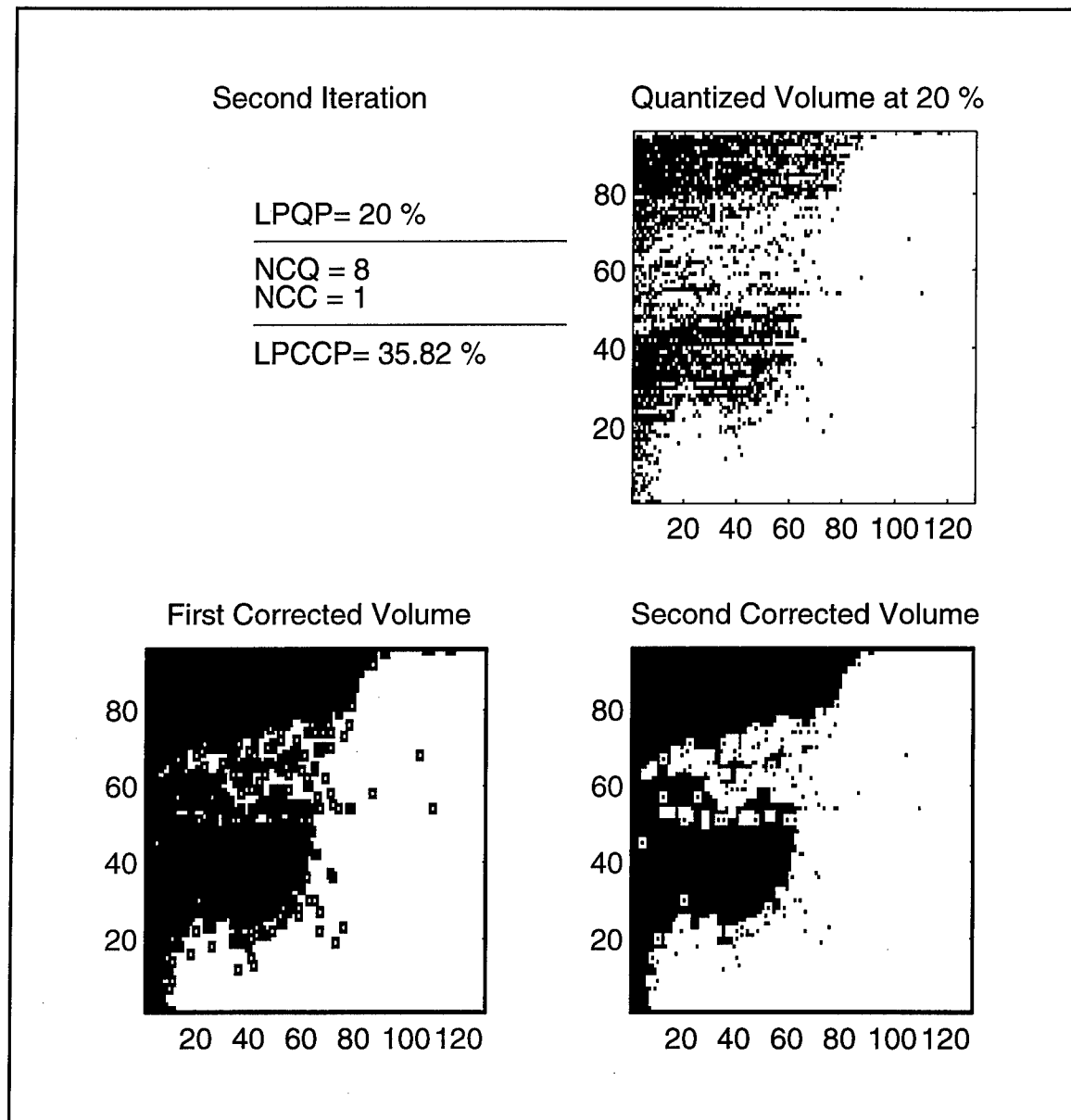
Figure 5.8 - <u>Step 2 of the Identification of LP and RPs</u>

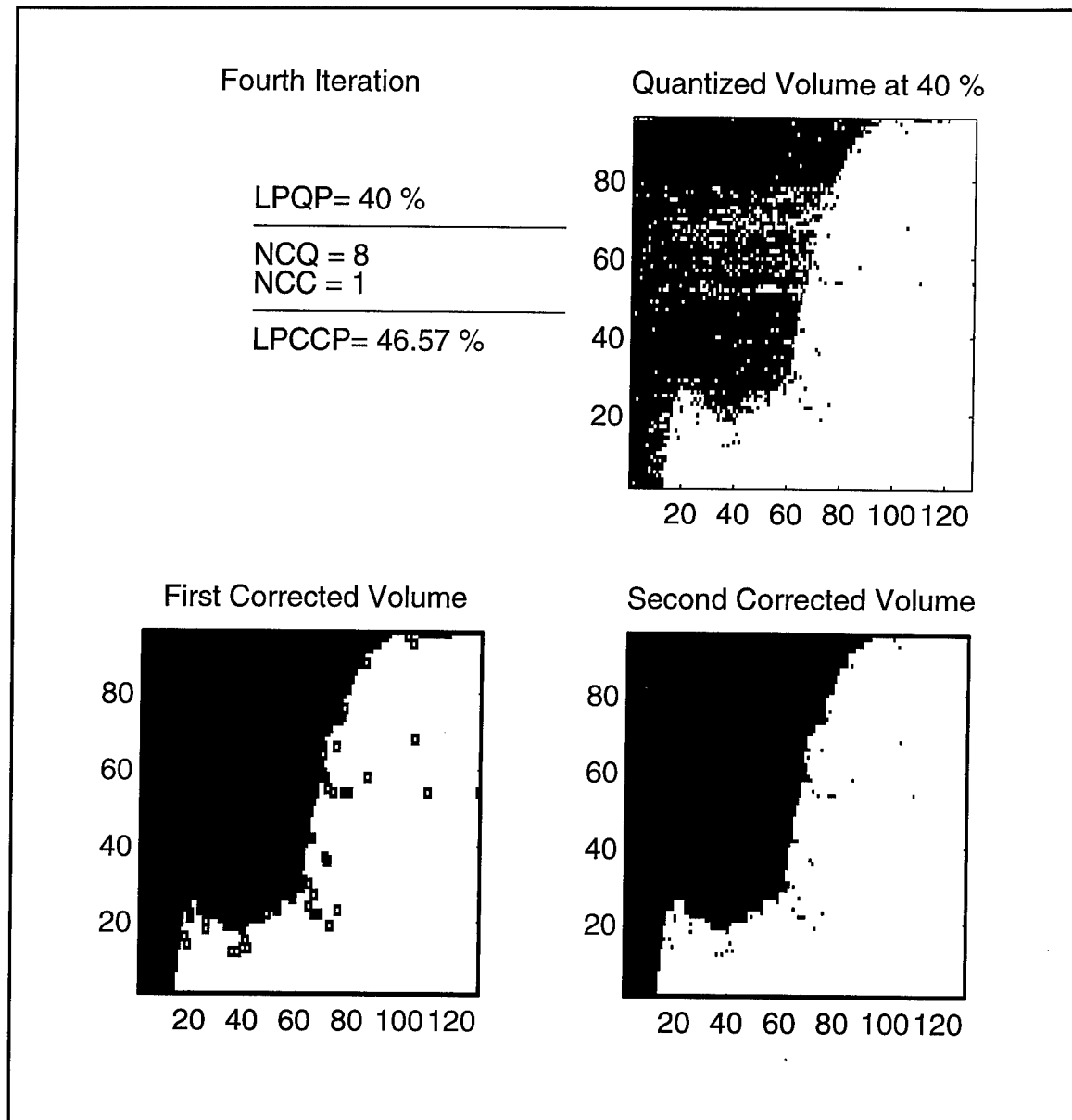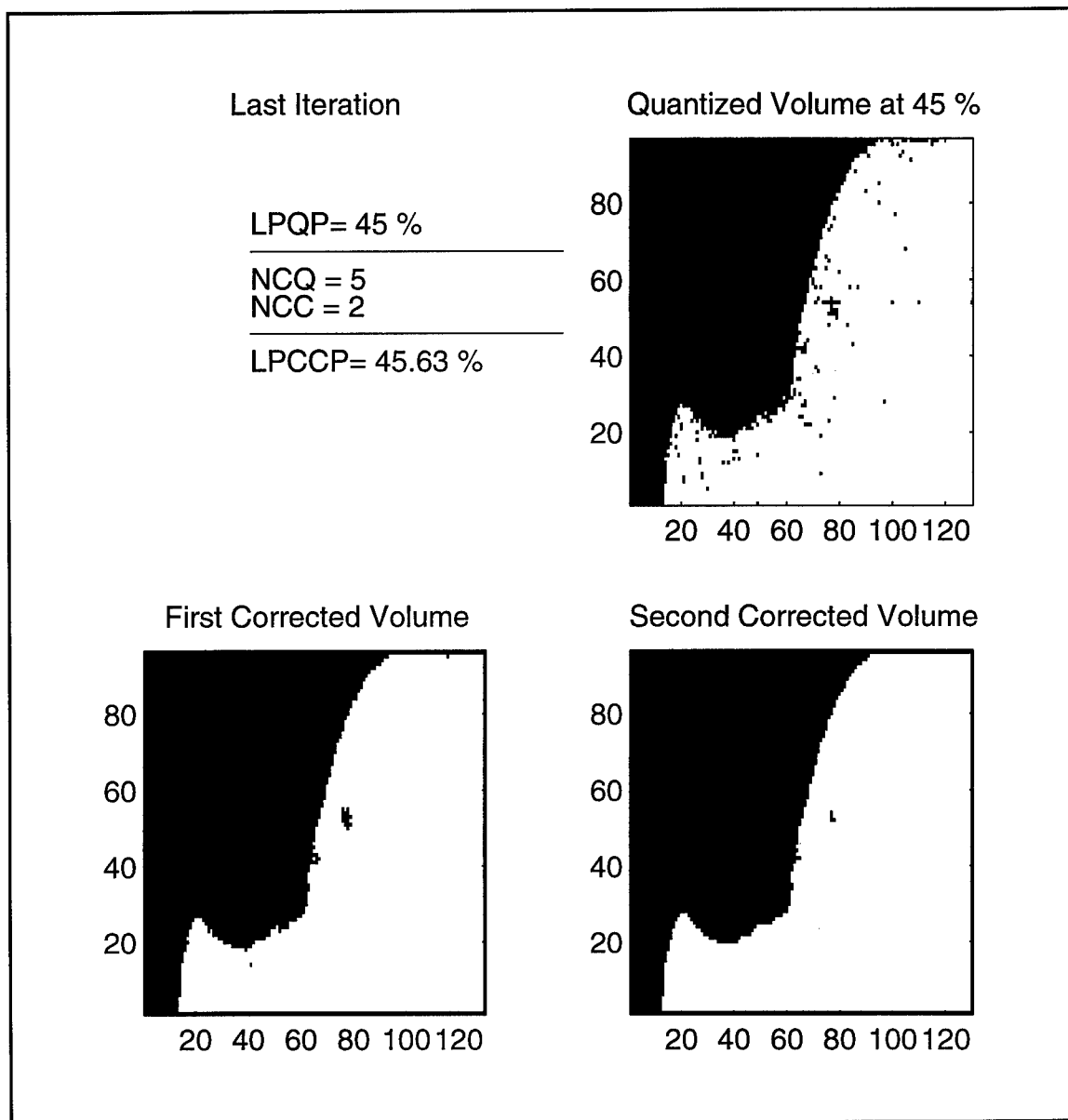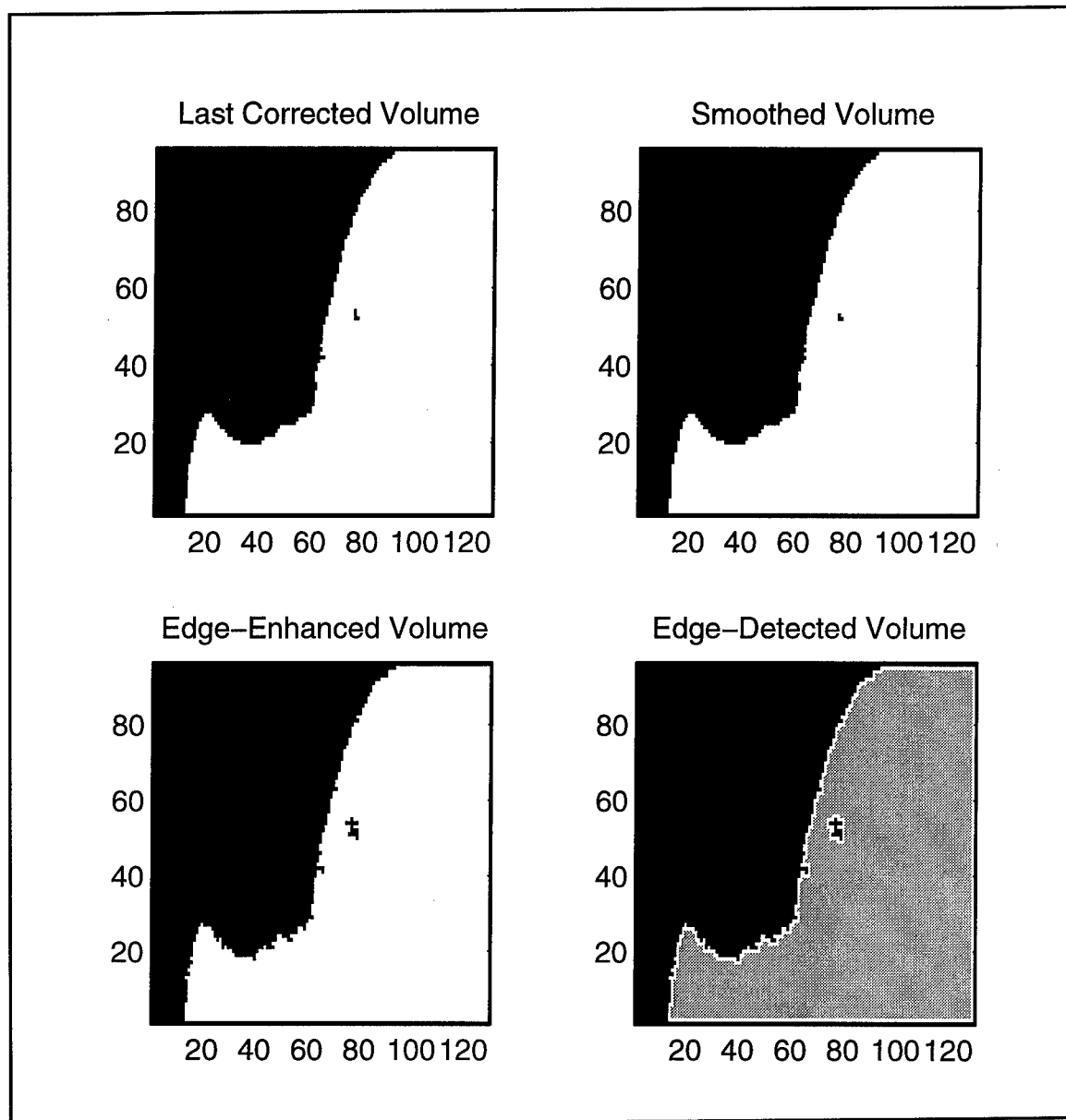**Figure 5.9 - <u>Step 3 of the Identification of LP and RPs</u>**

Fourth Iteration

LPQP= 40 %

NCQ = 8
NCC = 1

LPCCP= 46.57 %

Quantized Volume at 40 %

First Corrected Volume

Second Corrected Volume

Figure 5.10 - Step 4 of the Identification of LP and RPs

Figure 5.11 - <u>Step 7 of the Identification of LP and RPs</u>

Figure 5.12 - <u>Detection of Patch Edges Stage</u>

and 2 which contain 6680 and 5337 pixels, respectively. Furthermore, the values in Table 5.2 of the variance, mean and average power of patch 3 are closer to those of the lake than those of the land. This indicates that patch 3 is more related to the lake than the land and may be for example a small body of water. Next, the mapping procedure is used to investigate the presence of any subpatches in every previously detected patch (i.e., patches labeled 1, 2, and 3). In this case, no subpatches are detected and all three patches are declared as homogeneous.

5.5 - Statistical Procedure Stage

The statistical procedure is applied to every previously declared homogeneous patch in order to separate further between any existing contiguous subpatches that may have similar power levels but different data distributions. Following the steps developed in Chapter 3, the data of the scene is processed as follows:

Step 1

Test pixels and their respective tiles (sets of 100 reference pixels) are selected, spread throughout the patch, to be tested for Gaussianity using the goodness of fit test of the Ozturk algorithm. Note that the sets of 100 pixels are chosen to be disjoint, the closest to and belonging to the same patch as their respective test pixels. This results in the sets being shaped as 10x10 square tiles inside the patch and tiles tracking the shape of the boundary near the boundary of the patch. The result of this step is shown in Figure 5.13, where the Gaussian and non-Gaussian tiles are shaded in gray and white, respectively. Note that many non-Gaussian sub-regions exist in patches 1 and 2. Note also that patch 3 is not processed by the statistical procedure due to the fact that it is composed of only 15 pixels while a minimum of 100 pixels are needed for the Ozturk algorithm to result in a meaningful approximation for the distribution of the data.

Step 2

Once the Gaussian and non-Gaussian regions are determined in each patch, pixels with outliers are determined in the non-Gaussian declared tiles (sets of 100 pixels) and excised.
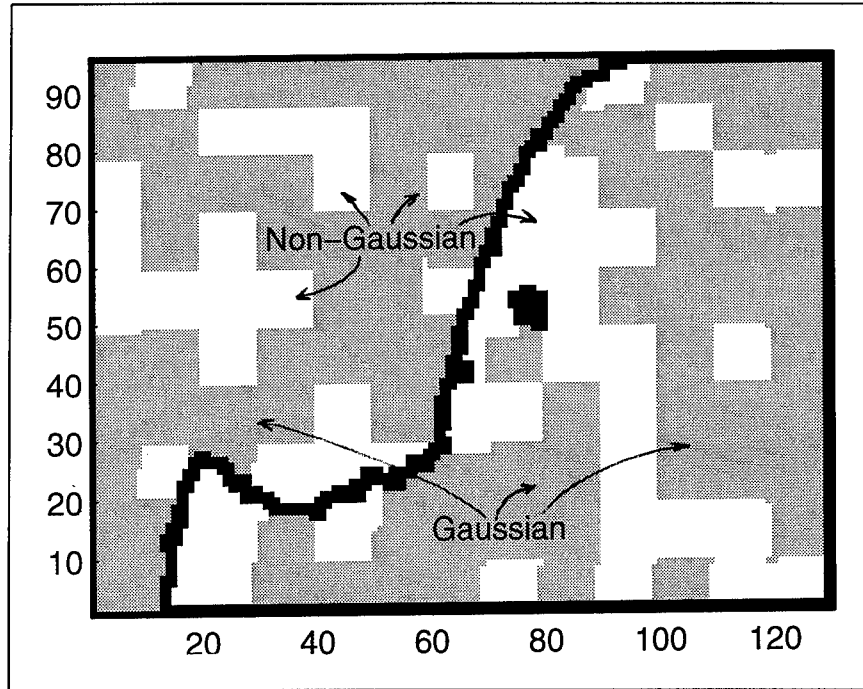
Figure 5.13 - <u>Scene After Step 1 of the Statistical Procedure</u>

Step 3

Step 3 of the statistical procedure proceeds by re-checking for Gaussianity the previously non-Gaussian declared tiles when their outliers are excised. The result is shown in Figure 5.14. Note that even though a lot of non-Gaussian previously declared tiles are now declared Gaussian, non-Gaussian tiles still exist particularly in the land region near the boundary with the lake. This is because, as stated previously, many non-homogeneous regions due mainly to trees and roads exist in that region causing discretes to appear in the data thus making it non-Gaussian. Figure 5.15 shows the location of outliers (black dots). Comparing Figures 5.15 and 5.4, note that outliers tend to have a physical significance and might represent string-like patches such as road-ways.

The results of the statistical procedure (steps 1 to 3) are summarized in Table 5.3. Note that:
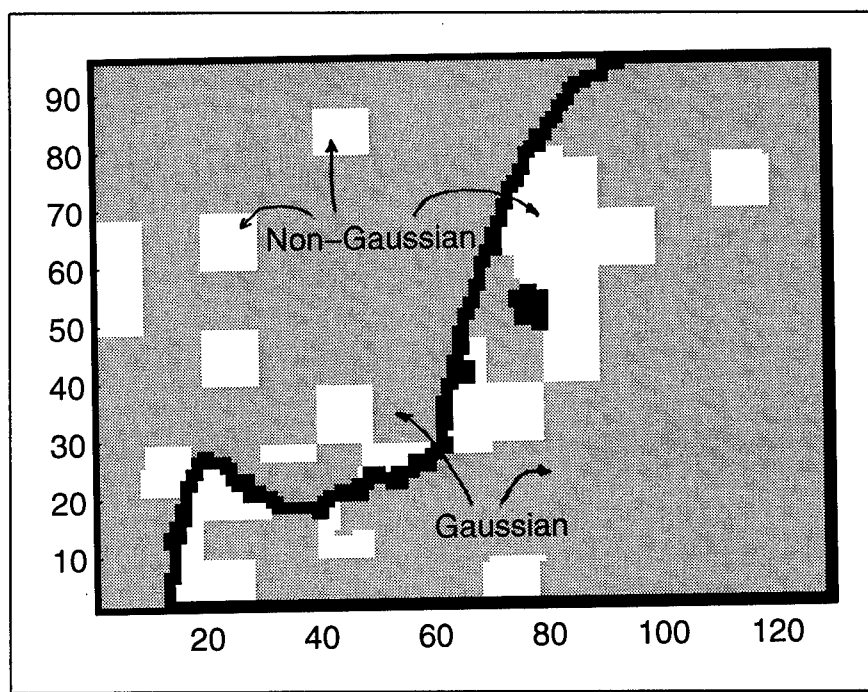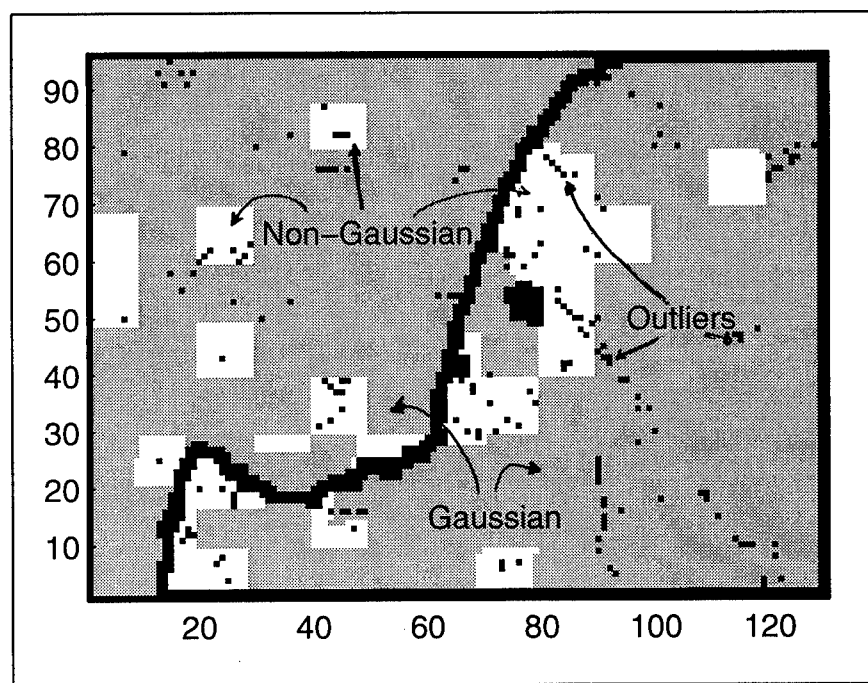
70

Figure 5.14 - Scene After Step 3 of the Statistical Procedure

Figure 5.15 - Scene After Step 3 of the Statistical Procedure and Location of Outliers

(1) the outliers occupy a small percentage of the scene (1.63%),

(2) Gaussian regions occupy the largest part of the scene, and

(3) non-Gaussian regions exist and occupy about 16% of the scene.

	Percentage of Gaussian Pixels (%)	Percentage of Non-Gaussian Pixels (%)	Percentage of Outliers (%)
Tiles with present outliers	61.81	38.19	-
Tiles with excised outliers	82.43	15.94	1.63

Table 5.3 - <u>Percentages of Gaussian and Non-Gaussian regions</u>

Step 4

In this step, the following procedure is followed,

1 - non-Gaussian tiles are numbered as shown in Figure 5.16.

2 - contiguous numbered tiles are then grouped in sets as shown in column 1 of Table 5.4.

3 - Using the Ozturk algorithm, the (U,V) coordinates of the locus end point is obtained for every tile.

4 - The average (Uav,Vav) coordinates is computed for every set of contiguous non-Gaussian tiles and the average approximating PDF corresponding to the pair (Uav,Vav) is obtained. The PDF type along with the first and second shape parameters are shown in columns 3, 4, and 5 of Table 5.4, respectively for every set of tiles. Note that a set can consist of a single tile if the tile is not contiguous to any other non-Gaussian tiles.

Tile Number	Fitting under same ellipse	PDF Type	1st Shape Parameter	2nd Shape Parameter
1	-	Beta	6.0	1.6
2	-	Beta	1.0	3.2
3	-	SU-J	1.2	-0.2
4, 5	Yes	Beta	1.0	3.2
6, 7, 8	No	Beta	2.0	3.2
9 to 16	Not all	Beta	1.0	3.2
17	-	Beta	1.0	0.8
18	-	Beta	2.0	0.8
19	-	Beta	2.0	1.6
20, 21	Yes	Beta	2.0	1.6
22	-	SU-J	1.2	-0.2
23	-	Beta	1.0	1.6

Table 5.4 - **PDFs to approximate the Different Subpatches**

5 - A check is made to ensure that all (U,V) pairs of every set fit withing the same confidence ellipse as (Uav,Vav). The result of this check is reported in column 2 of Table 5.4. Note that two sets, consisting of tiles numbered 6 to 8 and 9 to 16, do not pass the check.

6 - For those sets that do not pass the check above, the tiles are regrouped so that all (U,V) coordinates for each group of tiles can fit within the same ellipse. Each group of contiguous tiles form then a subpatch. Table 5.5 shows the result of this step. Note that column 1 displays the sets of contiguous tiles whereas column 2 allocates numbers to the subpatches obtained by the statistical procedure. Note also that each subpatch has its PDF

Tile Number	Sub-patch Number	Fitting under same ellipse	PDF Type	1st Shape Parameter	2nd Shape Parameter
1	1	-	Beta	6.0	1.6
2	2	-	Beta	1.0	3.2
3	3	-	SU-J	1.2	-0.2
4, 5	4	Yes	Beta	1.0	3.2
6	5	-	Beta	1.0	0.8
7	6	-	Beta	1.0	1.6
8	7	-	Lognormal	0.6	N/A
9 to 15	8	Yes	Beta	1.0	3.2
16	9		Beta	0.6	0.8
17	10	-	Beta	1.0	0.8
18	11	-	Beta	2.0	0.8
19	12	-	Beta	2.0	1.6
20, 21	13	Yes	Beta	2.0	1.6
22	14	-	SU-J	1.2	-0.2
23	15	-	Beta	1.0	1.6

Table 5.5 - <u>PDFs to approximate the Different Subpatches</u>

approximated as shown in columns 4 to 6 of Table 5.5.

At the end of the statistical procedure, subpatches within the lake and land regions are isolated. Outliers which represent tiny patches such as road ways are located and for every pixel in the scene, the PDF to approximate the data is readily known.
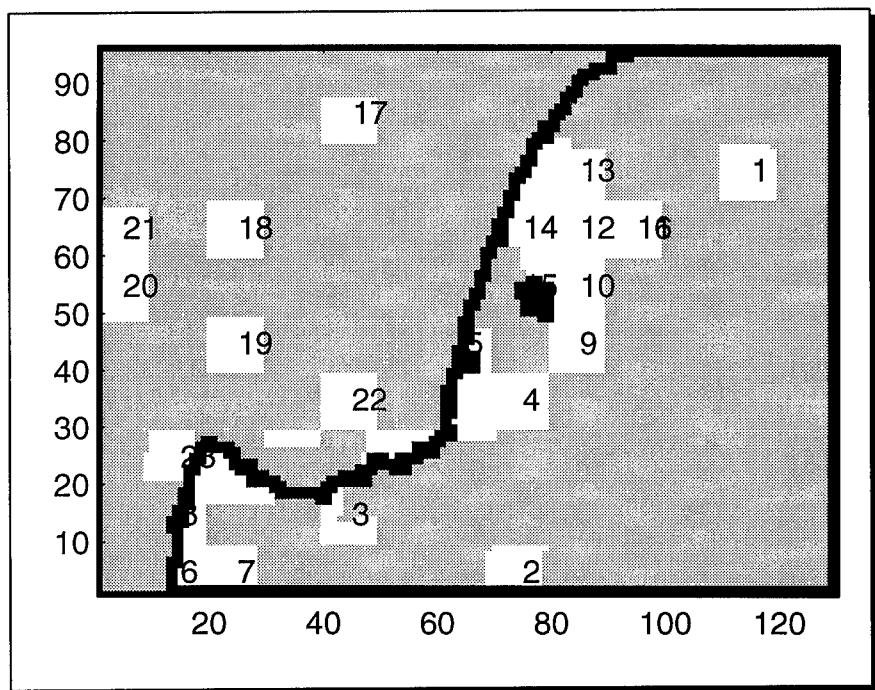
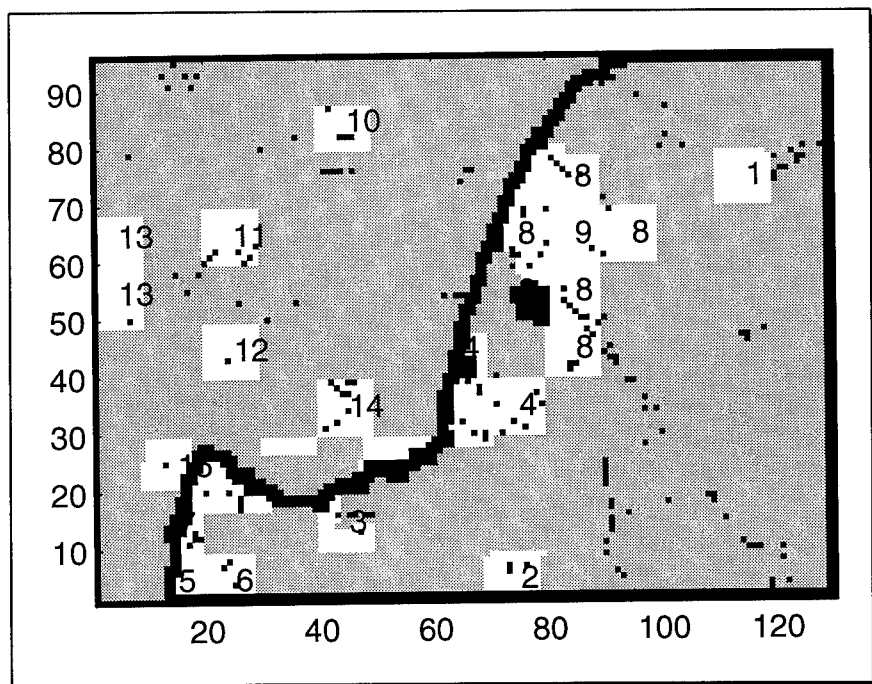**Figure 5.16 - <u>Numbering of the Non-Gaussian Tiles</u>**

Figure 5.17 - <u>Scene after the Statistical Procedure</u>

5.6 - Indexing Stage

The statistical procedure stage is followed by the indexing stage in which every pixel in the scene is assigned a set of descriptors to indicate, for each pixel, (1) its identification number which is the same as the identification number of the homogeneous patch or subpatch to which the pixel belongs, (2) whether the pixel is an inner pixel or edge pixel, (3) its type (Gaussian, non-Gaussian, or outlier), (4) for a non-Gaussian pixel, its best approximating PDF.

Up to this point, patches 1, 2, and 3 have been identified, outliers have been located in patches 1 and 2. Also, subpatches have been defined within every one of patches 1 and 2 and PDFs to approximate the different non-Gaussian subpatches are determined using the application chart mode of the Ozturk algorithm. All of this information is readily available and stored in the indexing stage.

5.7 - Conclusion

The processing of the real IR data by A'SCAPE has resulted during the mapping procedure in the partitioning of the scene into three main patches, lake, land, and a subpatch within the land. Then, the statistical procedure partitioned further the lake and land into main Gaussian patches and a total of fifteen non-Gaussian subpatches. Also, outliers representing tiny patches such as roads have been located and approximating PDFs have been determined for the non-Gaussian subpatches. Finally, if A'SCAPE is to be followed by a detection stage, all information pertaining to what type of detector should be used and what the parameters should be is readily available.

A'SCAPE Demo Package

6.1 - Introduction

A demo package is built in Matlab which describes in detail the different stages of A'SCAPE. The package has a friendly mouse driven graphical user interface (GUI) and consists of two main sections. The first section presents the detailed steps of the real IR data example of Chapter 5. The second section is subdivided into two subsections where in the first one a set of examples is presented which illustrate the need for A'SCAPE, whereas in the second, a detailed description is given for every stage of A'SCAPE. The views can be displayed manually or in an automated way as a slide-show. A movie is included which shows the steps in the mapping procedure. The next sub-sections describe in detail the different parts of the demo package.

As shown in Figure 6.1, note that the A'SCAPE demo consists of three windows referred to as the menu window (top right), plot window(top left), and command window (bottom left). These are described next.

6.2 - Menu Window

The menu window consists of a main menu and sub-menus formed each of a set of vertically aligned push buttons. Each button when pressed either generates another menu or plots in the Plot Window. The first menu to appear in the demo is referred to as the Main Menu. As shown in Figure 6.2, the first three buttons of the main menu are labeled Original Scene, Mapping Procedure, and Statistical Procedure, respectively. These buttons run the example of the real IR data scene treated in chapter 5. The fourth button, labeled A'SCAPE Tutorial, offers a tutorial on A'SCAPE. The fifth button, labeled About A'SCAPE, gives credit to the author and sponsors of the work. Finally, the button labeled end stops the demo and closes all windows.

All of the first four buttons generate sub-menus labeled as summarized in Table 6.1.

Figure 6.1 - A'SCAPE Demo,

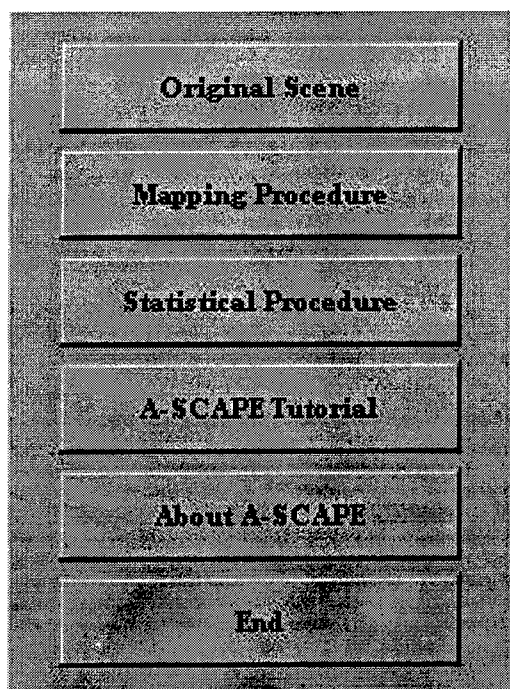**Menu Window (top right), Plot Window (top left), Command Window (bottom left)**

Figure 6.2 - A'SCAPE Demo: Main Menu

Main Menu Button Number	Effects
1	Sub-menu of Figure 6.3 labeled Original Scene Menu
2	Sub-menu of Figure 6.4 labeled Mapping Procedure Menu
3	Sub-menu of Figure 6.5 labeled Statistical Procedure Menu
4	Sub-menu of Figure 6.6 labeled A'SCAPE Tutorial Menu

Table 6.1 - **Effects of Buttons 1 to 4 of the Main Menu**

Effects of the buttons in each of the sub-menus, generated by buttons 1 to 4 of the main menu and shown in Figures 6.3 through 6.5, are summarized in tables 6.2 to 6.5, respectively.
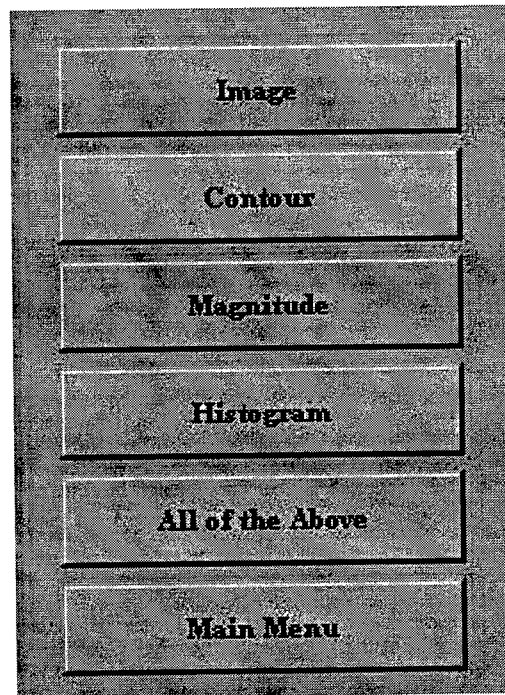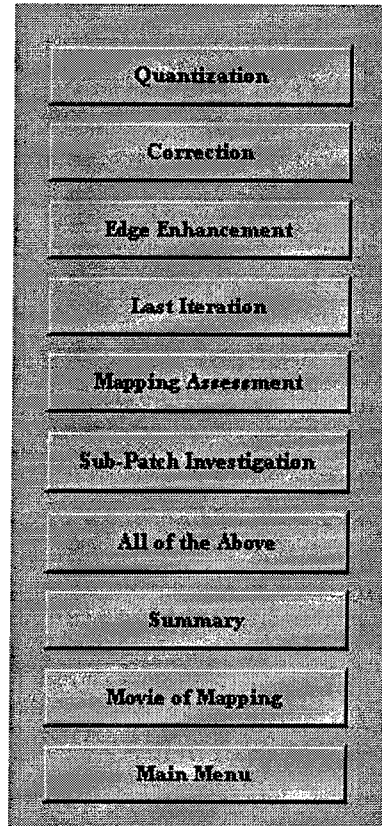

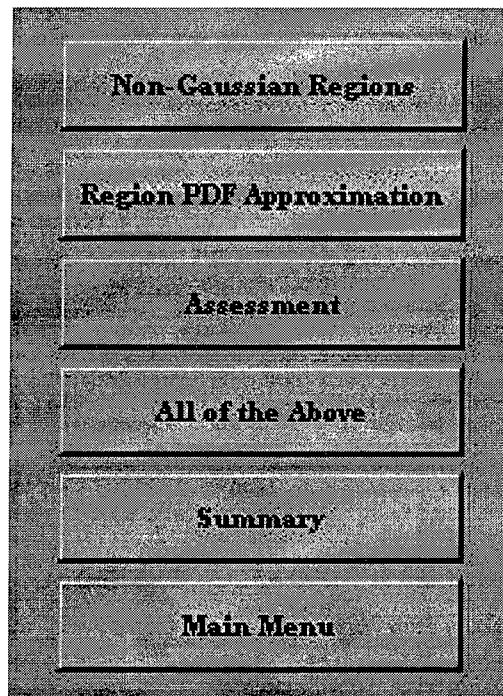Figure 6.3 - **Original Scene Menu**

Figure 6.4 - **Mapping Procedure Menu**
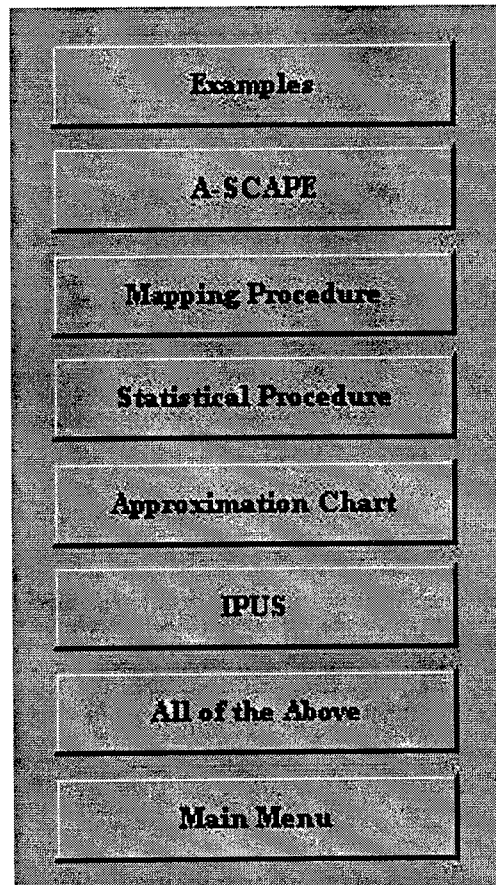

Figure 6.5 - **Statistical Procedure Menu**

Figure 6.6 - <u>A'SCAPE Tutorial Menu</u>

Original Scene Menu Button Label	Effects
Image	Displays the image of the original scene
Contour	Displays the contour plot of the original scene
Magnitude	Displays the 3 dimensional plot of the original scene
Histogram	Displays the histogram of the original scene
All of the Above	Displays all of the above in a single window
Main Menu	Closes Original Scene Menu and opens Main Menu

Table 6.2 - <u>Effects of Buttons in the Original Scene Menu</u>

Mapping Procedure Menu Button Label	Effects
Quantization	Displays the quantization steps
Correction	Displays the correction steps
Edge Enhancement	Displays the edge enhancement steps
Last Iteration	Displays the steps in the last iteration
Mapping Assessment	Displays the steps in the assessment of the Mapping Procedure
Sub-Patch Investigation	Displays the steps of the investigation of sub-patches
All of the Above	Displays all of the above in a sequential manner
Summary	Displays all of the above in a compact manner
Movie of Mapping	Displays the mapping procedure as a movie
Main Menu	Closes Mapping Procedure Menu and opens Main Menu

Table 6.3 - <u>Effects of Buttons in the Mapping Procedure Menu</u>

Statistical Procedure Menu Button Label	Effects
Non-Gaussian Regions	Displays the search for Gaussian and Non-Gaussian region steps
Region PDF Approximation	Displays the search for PDFs approximating Non-Gaussian region
Assessment	Assessment of the Statistical Procedure
All of the Above	Displays all of the above in a sequential manner
Summary	Displays all of the above in a compact manner
Main Menu	Closes Statistical Procedure Menu and opens Main Menu

Table 6.4 - <u>Effects of Buttons in the Mapping Procedure Menu</u>

A'SCAPE Tutorial Menu Button Label	Effects
Examples	Displays examples to justify the need for A'SCAPE
A'SCAPE	Displays block diagram of A'SCAPE
Mapping Procedure	Displays block diagram of the Mapping Procedure
Statistical Procedure	Displays block diagram of the Statistical Procedure
Approximation Chart	Displays an example of an approximation chart
IPUS	Displays a summary of the expert system main definitions
All of the Above	Displays all of the above in a sequential manner
Main Menu	Closes A'SCAPE Tutorial Menu and opens Main Menu

Table 6.5 - Effects of Buttons in the Mapping Procedure Menu

6.3 - Plot Window

All plots displayed by the different buttons are presented in the Plot Window. The plots are either full sized one plot at a time or compact four plots at a time.

6.4 - Command Window

When there is more than one plot to be displayed by a single button in any of the menus, buttons appear in the Command Window as shown in Figure 6.7 which consist of four push buttons and two sliders. Effects of the buttons are described in Table 6.6.
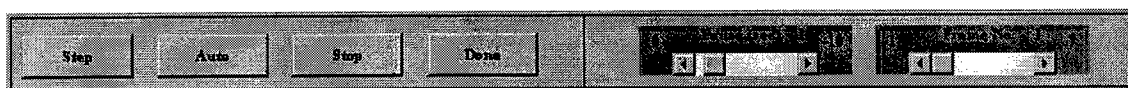

Figure 6.7 - Buttons in the Command Window

Command Window Button Label	Effects
Step	Displays frame by frame at the request of the user
Auto	Displays the frames automatically with a pause between the frames defined by the Pause slider
Stop	Stops the automatic display of the different frames
Done	Clears the command window and returns command to the menu window
Pause (slider)	Shows the pause in seconds between the frames being displayed automatically. Also enables choice of the pause in seconds when the slider is moved
Frame (slider)	Shows the number of the frame being displayed. Also, enables display of any desired frame when the slider is moved

Table 6.6 - <u>Effects of Buttons in the Mapping Procedure Menu</u>

Part II

Performance Analysis of Multichannel Parametric

Detection Algorithms

1

Introduction

In many signal processing techniques, the number of signals, say q, is assumed to be known a priori. This is needed for further processing of the received data vector. However, in practice, q is unknown and therefore needs to be estimated. This problem is often referred to in the literature as a "Detection Problem". For simplicity, we review the basis of the subspace approaches. This is needed for a good understanding of our work. Let the number of elements in an array of sensors be denoted by m, where m is assumed to be greater than q. The received signal vector can be expressed as

$$\underline{X} = A\underline{S} + \underline{N} \tag{1-1}$$

where $\underline{X}$ and $\underline{N}$ are (mx1) complex vectors, $\underline{S}$ is a (qx1) complex vector and A is an (mxq) complex matrix containing information about the signals of interest. $\underline{S}$ refers to the random signal vector while $\underline{N}$ is a noise vector. All signal and noise wave forms are assumed to be zero-mean random narrow band processes. Moreover, for simplicity, the additive noise vector is assumed to have a diagonal covariance matrix given by $\sigma_N^2 I_m$, where I_m is the (mxm) identity matrix. The case of correlated noise is considered in section 2.2.2. We also assume that the signal and noise vector are statistically independent. Note that the analysis considers the spatial domain only.

It follows from the signal model represented by Equation (1-1) and the above assumptions that the covariance matrix of $\underline{X}$ is given by

$$R_X = E\left[\underline{X} \bullet \underline{X}^H\right] = ASA^H + R_N = R_S + R_N = R_S + \sigma_N^2 I_m \tag{1-2}$$

where R_S is the covariance matrix of the signal vector $\underline{S}$ and H denotes the complex conjugate transpose operation. In an eigen-decomposition, note that the rank of the (mxm) matrix R_X is q. Therefore the smallest eigenvalues of R_X are identical and are equal to σ^2. If the eigenvalues of R_X are ordered in descending order with respect to their magnitudes, such that $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_q \geq \cdots \geq \lambda_m$, it follows that the q largest eigenvalues of R_X have magnitudes greater than σ^2. Specifically,

$$\lambda_i \geq \sigma_N^2 \; ; i=1, 2, ..., q \qquad\qquad (1\text{-}3)$$

and

$$\lambda_i = \sigma_N^2 \; ; i = (q+1), (q+2), ..., m. \qquad\qquad (1\text{-}4)$$

If the exact eigenvalues of the covariance matrix R_x of the received signal were available to us, the solution to the problem of detection (estimating q) would be straightforward in that we would simply have to count the multiplicity of the smallest eigenvalue. In practice, however, the covariance matrix R_x and its eigenvalues are unknown and they have to be estimated from a finite set of data samples. Due to the noisy nature of these estimates, we find that the exact multiplicity rule does not hold. Hence, this simple rule for determining the number of targets has to be dropped in favor of a more robust approach.

The first technique to be used in the detection problem was based on a statistical analysis of the received data. Various criteria are described in the literature for determining the number of targets. The two most commonly used are the Akaike information criterion (AIC) introduced by Akaike[1], which was first introduced in Direction Finding (DF) problems by Wax et al[2], and the Minimum Descriptive Length (MDL) criterion of Shwartz[3] and Rissanen[4].

In both approaches, the signals and m additive noise wave forms are assumed to be ergodic Gaussian processes. Let $\hat{R}_x$ denote the Maximum Likelihood (ML) estimate of the covariance matrix R_x. Also, let $\left\{ l_k \; ; k=1,2,\cdots,m, \; l_1 \geq l_2 \geq \cdots \geq l_m \right\}$ be the eigenvalues of $\hat{R}_x$. The cost function for both the AIC and MDL criterion takes on the form

[1] H. Akaike, "A New Nook at the Statistical Model Identification," IEEE trans. Automat. Contr., Vol. AC - 14, pp. 716-723, Dec. 1974.

[2] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-33, pp. 387-392, Apr. 1985.

[3] G. Schwartz, "Estimating the Dimension of a Model." Anna. Stat., Vol. 6, No. 22, pp. 461-464, 1978.

[4] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Descriptive Length," Ann. Stat., Vol. 11, No. 2, pp. 466-471, 1983.

$$J(k) = L(k) + p(k) \tag{1-5}$$

where the log-likelihood function, L(k) is given by

$$L(k) = N \cdot Ln \left\{ \frac{\left(\frac{1}{(m-k)} \sum_{i=k+1}^{m} l_i \right)^{m-k}}{\prod_{i=K=1}^{m} l_i} \right\}, \tag{1-6}$$

with N being the number of time samples (snapshots) used in evaluating $\hat{R}_X$, and the penalty function, p(k), is given by

$$p(k) = \alpha(N)[k(2m-k)], \tag{1-7}$$

with $\alpha(N)$, the penalty coefficient, being a function of the number of samples.

The choice of $\alpha(N)$ depends on the criterion used. For the AIC, $\alpha(N) = 1$ whereas for the MDL criterion, $\alpha(N) = (1/2)\ln(N)$. The number of signals $\hat{q}$ is then determined as that value of k for which J(k) is minimized.

Let H_q denote the hypothesis that the true number of targets is q. Then, the probability of underestimating and overestimating the number of targets, given H_q, are

$$P_M = P(\hat{q} < a / H_q) \tag{1-8}$$

$$P_F = P(\hat{q} > q / H_q) \tag{1-9}$$

P_M and P_F are called the miss and false alarm probabilities, respectively. It has been shown[5] that the probability of missing for both the AIC and MDL criterion can be reduced to zero by either increasing the signal-to-noise ratio (SNR) or by increasing the number of samples N. However, for both these parameters, the decrease is much faster for AIC than for MDL. This indicates that the AIC is more efficient in reducing the probability of missing than

---

[5] Q. Zhang, K. Wang, Y. Yin, J. Reilly, "Statistical Analysis of the Performance of Information Theoretic Criteria in the Detection of the Number of Signals in Array Processing," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-37, pp. 1557-1567, Oct. 1989.

is the MDL criterion. On the other hand, for a moderate number of samples the probability of false alarm using the MDL criterion approaches zero with increasing values of N, whereas that for the AIC remains constant. This shows that the MDL criterion is more efficient in reducing the probability of false alarm than is the AIC. Under low SNR's, both the AIC and the MDL necessitate a large number of samples, which may be very hard to meet in practice.

Research is needed to determine whether the performance of the information theoretic criteria can be improved for specific applications by selecting other functions for $\alpha(N)$ and/or log-likelihood functions $L(k)$.

Another technique[6], based on the AIC and the MDL, has been applied on the singular values of the covariance matrix instead of the eigenvalues of the covariance matrix. The reason is that there is a direct relationship between the eigenvalues of the covariance matrix and its singular values. The advantage of this technique over the previous one is that it benefits from the numerical efficiency and stability of the SVD technique.

Recently, a new procedure[7] for the detection of the number of targets was introduced, within the framework of model selection, and established the strong consistency of the procedure. This procedure is referred to as the Efficient Detection Criterion (EDC) procedure. According to this procedure, q is estimated with $\hat{q}$ where

$$EDC(\hat{q}) = \min\{EDC(0), \ldots, EDC(m-1)\} \qquad (1\text{-}10)$$

where

$$EDC(k) = -2Ln(k) + v(k,m)C(N) \qquad (1\text{-}11)$$

and C(N) satisfies the following conditions:

$$i) \quad \lim_{N \to \infty}[C(N)/N] = 0 \qquad (1\text{-}12)$$

[6] V. Shamirian and S.. B. Kesler, "Detection of Signals by SVD-Based Information Theoretic Criteria," Proc. 30-th Midwest Symposium on Circuits and Systems, pp. 567-570, Syracuse, NY 1987.

[7] L. C. Zhao et al., "Remarks on Certain Criteria for Detection of Number of Signals," IEEE Trans. ASSP, Vol. ASSP-35, No. 11, pp. 129-132, Feb. 1987.

$$\text{ii)} \quad \lim_{N \to \infty} \left\{ C(N)/Ln[Ln(N)] \right\} = 0 . \qquad (1\text{-}13)$$

A very general and elegant method[8] to estimate the number of targets was recently introduced. This method depends on choosing a function f defined on all finite sequences, called r-regular functions. The previous methods used only 1-regular functions (r = 1). It was shown that actually this is the worst case. It was eventually shown that as r increases, the convergence is even faster.

From the above discussion, we notice that most techniques are non-parametric. It was our belief that parametric techniques are better suited for this type of detection since they have more degrees of freedom and eventually will prove very efficient.

Our interest in the parametric techniques and LeCadre's technique[9] in particular is two fold. First, we evaluated and studied its performance against several parameters such as order determination of the AR model, number of incoming signals (targets), signal-to-noise ratios, and the presence or absence of clutter. We also studied the effects of a limited data sequence, computational complexity, algorithm convergence and implementation feasibility. This step could be referred to as a complete assessment of the proposed technique. In the second phase, we mainly restrict ourselves to the software development of appropriate detection and estimation algorithms for insertion in the Rome Lab Space Time Signal Processing (RLSTAP) software.

---

[8] Y. Q. Yin and P. K. Krishnaiah, "On Some Non-Parametric Methods for Detection of the Number of Signals," IEEE Trans. ASSP, Vol. ASSP-35, No. 11, pp. 1523-1538, Nov. 1988.

[9] J. P. LeCadre, "Parametric Methods for Spatial Signal Processing in the Presence of Unknown Colored Noise Fields," IEEE Trans. ASPP, Vol. 37, No. 7, pp. 965-983, July 1989.

2

LeCadre's Technique

In this section, we will derive the detection algorithm defined by LeCadre and provide insights into the models used in the derivation.

2.1 - Introduction

Let $\underline{X}_i$ be the i-th stationary, m-dimensional random vector constituted by the narrow-band outputs of an array sampling a homogeneous random field, where m is the number of sensors. $\underline{X}_i$ can be written as

$$\underline{X}_i = \begin{bmatrix} x_1(i) & x_2(i) & ... & x_m(i) \end{bmatrix}^T . \tag{2-1}$$

Let R_X be the covariance matrix of the outputs vector $\underline{X}_i$. R_X can be written as

$$R_X = E\begin{bmatrix} \underline{X}_i \bullet \underline{X}_i^H \end{bmatrix} . \tag{2-2}$$

It can easily be shown that R_X can be expressed as

$$R_X = R_S + R_N , \tag{2-3}$$

where R_S and R_N are the covariance matrices of the sources and noise, respectively.

The objective is to obtain an accurate estimation of R_N, based on the availability of an estimate of the covariance matrix $\hat{R}_X$. The method relies on 3 facts:

1. The likelihood functional may be expressed as a function of the eigenvalues of the whitened covariance matrix of the outputs.

2. The inverse N^{-1} of the noise covariance matrix admits an explicit formulation in terms of the AR(MA) coefficients of the noise model.

3. The derivatives of the likelihood functional may be computed using (1) and (2) and classical results for perturbations of eigenvalues.

2.2 - Calculation of the Likelihood Functional

The derivation of the likelihood functional depends upon which type of interference (i.e.; thermal white noise only or correlated clutter plus additive white noise) is being used. The first case will be devoted to the white noise interference and the second case will be devoted to the non-white case.

2.2.1 - White Noise case

In this case, we assume that R_N is of the form $R_N = \lambda_N I_m$, where λ_N is a positive constant and I_m is the (mxm) identity matrix. Moreover, assume that the number of sources is denoted by q. Let $\{\underline{X}_1, \underline{X}_2, \cdots, \underline{X}_N\}$ be a sequence of N independent complex Gaussian vectors with covariance matrix R. Note that $\hat{R}$ is given by

$$\hat{R}_x = \frac{1}{N} \sum_{i=1}^{N} \underline{X}_i \bullet \underline{X}_i^H , \qquad (2\text{-}4)$$

where H denotes complex conjugate transpose.

Given R_S and λ_N, the pdf of any spatial vector snapshot $\underline{X}_i$ is given by

$$f_{\underline{X}_i}\left(\underline{x}/\lambda_N, R_S\right) = \frac{1}{(\pi)^m} \frac{1}{\det(R_x)} \exp\left\{-\underline{x}_i^H R_x^{-1} \underline{x}_i\right\}, \qquad (2\text{-}5)$$

where independence was assumed from sensor to sensor and over the sequence of N samples. The conditional density of the received sequence is then given by

$$f_{\underline{X}}\left(\underline{x}/\lambda_N, R_S\right) = \prod_{i=1}^{N} f_{\underline{X}_i}\left(\underline{x}_i/\lambda_N, R_S\right) = \frac{1}{(\pi)^{mN}} \frac{1}{\left[\det(R_x)\right]^N} \exp\left\{-\sum_{i=1}^{N} \underline{x}_i^H R_x^{-1} \underline{x}_i\right\}. \qquad (2\text{-}6)$$

To simplify this expression, we study the following example. Consider the case where

$$\underline{x}_i = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \text{ and let } R_x^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Note that $\displaystyle\sum_{i=1}^{N} \underline{x}_i^H R_x^{-1} \underline{x}_i$ is equal to

$$\sum_{i=1}^{N} \underline{x}_i^H R_x^{-1} \underline{x}_i = \sum_{i=1}^{N} \begin{bmatrix} x_{i1}^* & x_{i2}^* \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} = \sum_{i=1}^{N} a x_{i1} x_{i1}^* + b x_{i2} x_{i1}^* + c x_{i1} x_{i2}^* + d x_{i2} x_{i2}^* . \qquad (2\text{-}7)$$

On the other hand, note that

$$N \hat{R}_x = \sum_{i=1}^{N} \underline{X}_i \bullet \underline{X}_i^H = \sum_{i=1}^{N} \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \begin{bmatrix} x_{i1}^* & x_{i2}^* \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} x_{i1} x_{i1}^* & x_{i1} x_{i2}^* \\ x_{i2} x_{i1}^* & x_{i2} x_{i2}^* \end{bmatrix}. \qquad (2\text{-}8)$$

Therefore the product $N \hat{R}_x R_x^{-1}$ is given by

$$N \hat{R}_x R_x^{-1} = \sum_{i=1}^{N} \begin{bmatrix} x_{i1} x_{i1}^* & x_{i1} x_{i2}^* \\ x_{i2} x_{i1}^* & x_{i2} x_{i2}^* \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} a x_{i1} x_{i1}^* + c x_{i1} x_{i2}^* & b x_{i1} x_{i1}^* + d x_{i1} x_{i2}^* \\ a x_{i2} x_{i1}^* + c x_{i2} x_{i2}^* & b x_{i2} x_{i1}^* + d x_{i2} x_{i2}^* \end{bmatrix}. \qquad (2\text{-}9)$$

Recall that the trace of a matrix A is defined as the sum of its diagonal elements; i.e., $\mathrm{tr}(A) = \sum a_{ii}$. Note in this case that the trace of $N \hat{R}_x R_x^{-1}$ is given by

$$\mathrm{tr}\left(N \hat{R}_x R_x^{-1}\right) = N \mathrm{tr}\left(\hat{R}_x R_x^{-1}\right) = \sum_{i=1}^{N} a x_{i1} x_{i1}^* + c x_{i1} x_{i2}^* + b x_{i2} x_{i1}^* + d x_{i2} x_{i2}^* . \qquad (2\text{-}10)$$

Note that Equations (2-7) and (2-10) are identical. Hence, Equation (2.2.1-3) can be written as

$$f_{\underline{X}}\left(\underline{x}/\lambda_N, R_s\right) = \frac{1}{(\pi)^{Nm}} \frac{1}{[\det(R_x)]^N} \exp\left\{-N \mathrm{tr}\left(\hat{R}_x R_x^{-1}\right)\right\}. \qquad (2\text{-}11)$$

The next step is to find the value of λ_N and R_s which maximize the likelihood functional. Note that $\mathrm{Ln}\left[f_{\underline{X}}\left(\underline{x}/\lambda_N, R_s\right)\right]$ is given by

$$\mathrm{Ln}\left[f_{\underline{X}}\left(\underline{x}/\lambda_N, R_s\right)\right] = -N\left\{m\mathrm{Ln}(\pi) + \mathrm{Ln}\left[\det(R_x)\right] + \mathrm{tr}\left(\hat{R}_x R_x^{-1}\right)\right\}. \qquad (2\text{-}12)$$

It can be shown[10] that maximizing this function with respect to λ_N results in the following equation

---

[10] T. W. Anderson, "Asymptotic Theory for Principal Component Analysis," <u>Ann. J. Math. Stat.</u>, Vol. 34, pp. 122-148, 1963.

$$\max\left\{Ln\left[f_{\underline{x}}\left(\underline{x}/\lambda_N,R_S\right)\right]\right\}=N\left\{-mLn(\pi)-Ln\left[\det(R_x)\right]+\sum_{i=q+1}^{m}Ln\left(\hat{\lambda}_i\right)-(m-q)Ln\left[\frac{1}{m-q}\sum_{i=q+1}^{m}\hat{\lambda}_i\right]\right\}$$

(2-13)

Let C be the following constant

$$C=-mLn(\pi)-Ln\left[\det(R_x)\right],\qquad\qquad(2\text{-}14)$$

then, the above equation can be expressed as

$$\max\left\{Ln\left[f_{\underline{x}}\left(\underline{x}/\lambda_N,R_S\right)\right]\right\}=N\left\{C+(m-q)\frac{1}{(m-q)}Ln\left[\prod_{i=q+1}^{m}\left(\hat{\lambda}_i\right)\right]-(m-q)Ln\left[\frac{1}{m-q}\sum_{i=q+1}^{m}\hat{\lambda}_i\right]\right\}.$$

(2-15)

This last equation can also be written as

$$\max\left\{Ln\left[f_{\underline{x}}\left(\underline{x}/\lambda_N,R_S\right)\right]\right\}=N\left\{C+(m-q)Ln\left[\prod_{i=q+1}^{m}\left(\hat{\lambda}_i\right)\right]^{\left(\frac{1}{m-q}\right)}-(m-q)Ln\left[\frac{1}{m-q}\sum_{i=q+1}^{m}\hat{\lambda}_i\right]\right\}.$$

(2-16)

Therefore, it can easily be seen that maximizing the likelihood amounts to maximizing the functional

$$L_q\left(\lambda_N,R_S\right)=(m-q)Ln\left[ge(q)\right]-(m-q)Ln\left[ar(q)\right],\qquad(2\text{-}17)$$

where ge(q) and ar(q) are given by

$$ge(q)=\left[\prod_{i=q+1}^{m}\left(\hat{\lambda}_i\right)\right]^{\left(\frac{1}{m-q}\right)}\qquad\qquad(2\text{-}18)$$

and

$$ar(q)=\frac{1}{m-q}\sum_{i=q+1}^{m}\hat{\lambda}_i\ .\qquad\qquad(2\text{-}19)$$

94

ge(q) and ar(q) are referred to as the geometric and arithmetic means, respectively. Note also that in the previous derivation, we have assumed that the eigenvalues of $\hat{R}_x$ are ordered; i.e.,

$$\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_m. \tag{2-20}$$

We now consider the case of unknown correlated noise.

2.2.2 - Unknown Correlated Noise

In this case, the likelihood functional is given by

$$L_q(R_N, R_S) = -Ln[det(R_x)] - tr(R_x^{-1}\hat{R}_x). \tag{2-21}$$

Note that R_N is a positive definite matrix, therefore it admits a Cholesky decomposition in the form of two triangular factors (matrices). Let R_N be given by $R_N = L.L^H$. Consider then the transformed matrix of the exact covariances (after application of L); i.e.,

$$R_w = L^{-1}RL^{-H} = L^{-1}(R_S + \lambda_N R_N)L^{-H} = L^{-1}(R_S + \lambda_N LL^H)L^{-H}, \tag{2-22}$$

which can also be written as

$$R_w = L^{-1}R_S L^{-H} + \lambda_N L^{-1}LL^H L^{-H} = L^{-1}R_S L^{-H} + \lambda_N I_m. \tag{2-23}$$

Furthermore, let S_w be the transformed source's matrix; i.e., $S_w = L^{-1}R_S L^{-H}$. Then the factor $tr(R_x^{-1}\hat{R}_x)$ becomes

$$tr(R_x^{-1}\hat{R}_x) = tr[(L^{-H}R_w^{-1}L^{-1})(L\hat{R}_w L^H)] = tr(L^{-H}R_w^{-1}\hat{R}_w L^H). \tag{2-24}$$

For simplicity, let A and B be the following matrices, $A = L^{-H}R_w^{-1}$ and $B = \hat{R}_w L^H$. Also, recall that given two matrices A and B, it is known that tr(AB)= tr(BA). Therefore Equation (2-24) becomes

$$tr(R_x^{-1}\hat{R}_x) = tr(L^{-H}R_w^{-1}\hat{R}_w L^H) = tr(\hat{R}_w L^H L^{-H}R_w^{-1}) = tr(\hat{R}_w R_w^{-1}) = tr(R_w^{-1}\hat{R}_w). \tag{2-25}$$

Note also that the determinant of R_x, det(R_x), can be written as

95

$$\det\left(R_x\right) = \det\left(LR_wL^H\right) = \det(L) \bullet \det(L^H) \bullet \det(R_w) = \det(R_N) \bullet \det(R_w), \qquad (2\text{-}26)$$

since $R_N = L.L^H$ and $\det\left(R_N\right) = \det(L) \bullet \det(L^H)$. Therefore $Ln[\det(R_x)]$ is given by

$$Ln[\det\left(R_x\right)] = Ln[\det(R_N)] + Ln[\det(R_w)]. \qquad (2\text{-}27)$$

Thus, Equation (2-21) becomes

$$L_q\left(R_N, R_S\right) = -Ln\left[\det\left(R_w\right)\right] - tr\left(R_w^{-1}\hat{R}_w\right) - Ln[\det(R_N)]. \qquad (2\text{-}28)$$

Let $L_w\left(q\right)$ be the quantity

$$L_w\left(q\right) = -Ln\left[\det\left(R_w\right)\right] - tr\left(R_w^{-1}\hat{R}_w\right), \qquad (2\text{-}29)$$

then Equation (2-28) becomes

$$L_q\left(R_N, R_S\right) = L_w\left(q\right) - Ln[\det(R_N)]. \qquad (2\text{-}30)$$

Note also that the rank of the matrix S_w is given by

$$rank\left(S_w\right) = rank(S), with R_w = S_w + \lambda_N I_m. \qquad (2\text{-}31)$$

Thus, the maximization of $L_q\left(R_N, R_S\right)$ is the same as the maximization of $L_w\left(q\right)$, relative to the source's subspace. This question was raised before and the solution is known to be given by

$$\max_S L_w\left(q\right) = -(m-q)Ln\left[ar_w\left(q\right)\right] + (m-q)Ln\left[ge_w\left(q\right)\right] - Ln\left[\det\left(\hat{R}_w\right)\right] + C_1, \qquad (2\text{-}32)$$

where $ar_w\left(q\right)$ and $ge_w\left(q\right)$ are the arithmetic and geometric means of the $\left(m-q\right)$ lowest eigenvalues of $\hat{R}_w$, respectively and C_1 is a constant given by $C_1 = -mLn(\pi)$. Then, using Equation (2-30), it can easily be seen that

$$\max_{R_S, \lambda_N} L_q\left(R_N, R_S\right) = -Ln[\det(R_N)] - (m-q)Ln\left[ar_w\left(q\right)\right] + (m-q)Ln\left[ge_w\left(q\right)\right] - Ln\left[\det\left(\hat{R}_w\right)\right] + C_1$$
$$(2\text{-}33)$$

However, recall from Equation (2-27) that

$$-\text{Ln}[\det(\hat{R}_w)] = -\text{Ln}[\det(\hat{R}_x)] + \text{Ln}[\det(R_N)], \qquad (2\text{-}34)$$

which implies that Equation (2-33) becomes

$$\max_{R_S, \lambda_N} L_q(R_N, R_S) = -(m-q)\text{Ln}[ar_w(q)] + (m-q)\text{Ln}[ge_w(q)] - \text{Ln}[\det(\hat{R}_x)] + C_1. \qquad (2\text{-}35)$$

Note in this last equation that the last two terms do not depend on R_N or R_S. Therefore in the remainder of this report, we will only consider the following expression

$$L_q(R_N) = -(m-q)\text{Ln}[ar_w(q)] + (m-q)\text{Ln}[ge_w(q)]. \qquad (2\text{-}36)$$

The problem now is how to maximize $L_q(R_N)$ (for a given number of sources q), relative to parameters defining the noise matrix R_N. This of course will depend on the noise parameterization, which is discussed next.

2.3 - Noise Parameterization

Consider first the case of an AutoRegressive (AR) modeling of the noise. This means that noise received on a sensor can be "predicted" from noise received on other sensors. This assumption applies for the general case that we are considering, since usually we are dealing with an array of sensors. We will show later, that actually all noise may be described by AR(MA) modeling. Assuming an AR type modeling, it can be shown (Gohberg)[11] that the inverse of the covariance matrix is given by

$$R_N^{-1} = \frac{1}{\sigma^2}\left(A_1 . A_1^t - A_3 . A_3^t\right), \qquad (2\text{-}37)$$

where A_1 and A_3 are two triangular (mxm) Toeplitz matrices given by

---

[11] T. Kailath et. al., "Inverse of Toeplitz Operators, Innovations and Orthogonal Polynomials," SIAM Rev., Vol. 20, pp. 106-119, Jan. 1978.

$$A_1 = \begin{bmatrix} 1 & & & & \\ a_1 & & \underline{0} & & \\ \vdots & & & & \\ a_p & & & & \\ \underline{0} & a_p & \cdots & a_1 & 1 \end{bmatrix} \quad \text{and} \quad A_3 = \begin{bmatrix} & & \underline{0} & & \\ a_p & & \underline{0} & & \\ \vdots & & & & \underline{0} \\ a_2 & & & & \\ a_1 & a_2 & \cdots & a_p & \end{bmatrix}. \tag{2-38}$$

Note that this formula is valid for any real stationary AR noise of order p with coefficients $a_i ; i = 1, 2, \cdots, p$ and σ^2 is the input driving noise power. Define the matrix Z^i, where

$$Z^i(j,k) = \begin{cases} 1 & \text{if} \quad (j-k) = i \; ; \; 1 \le j \le m \\ 0 & \text{else} \quad ; \; 1 \le k \le m \end{cases}. \tag{2-39}$$

Notice that this matrix has the property that $Z^{i+1} = Z^i Z$, where Z^0 is the identity matrix. Then the matrices A_1 and A_3 can be written as

$$A_1 = \sum_{i=0}^{p} a_i Z^i \quad \text{and} \quad A_3 = \sum_{i=0}^{p} a_i Z^{m-i}, \tag{2-40}$$

where $a_0 = 1$, by convention.

In the case of an MA process, unfortunately, there is no explicit formulation of the inverse of the noise covariance matrix. However the following expression may be used

$$R_N = \left(\sum_{i=0}^{p} b_i Y_i \right) \left(\sum_{i=0}^{p} b_i Y_i \right)^t, \tag{2-41}$$

where t denotes transpose and the $b_i ; i = 1, 2, \cdots, p$ are the MA coefficients and $\{ Y_i \}$ are rectangular (mx2m) matrices defined by

$$Y_i(j,k) = \begin{cases} 1 & \text{if} \quad (k-j) = i \\ 0 & \text{else} \end{cases}. \tag{2-42}$$

By combining Equations (2-38) and (2-41), we obtain an expression for the parameterization of an ARMA process

$$R_N = \left(\sum_{i=0}^{p} b_i Y_i \right) R_{AR} \left(\sum_{i=0}^{p} b_i Y_i \right)^t, \tag{2-43}$$

98

where R_{AR} is the extended (2mx2m) matrix of the AR process as defined in Equation (2-37), but in dimension 2m.

The noise model used previously assumes that the noise is equal in each sensor and moreover that it has equal power. In the case where this assumption is not valid, the following noise model is better suited

$$X_m + a_1 X_{m-1} + a_p X_{m-p} = \beta_m w_n,$$ (2-44)

where w_m is a sequence of independent white noise and β_m is an additional parameter which has to be optimized.

There are several types of parameterizations for the noise covariance matrices, however, the ones which we consider have the advantage of defining a parameterization of R_N (or R_N^{-1}) with a small number of parameters.

In the next section we discuss ways of maximizing the likelihood functional in the case of an AR model.

2.4 - Maximization of the likelihood functional in the case of a real AR model

The maximization of L_q requires, in general, an iterative technique. A general form of the gradient algorithm is given by

$$\underline{A}_{k+1} = \underline{A}_k - \rho_k \underline{G}_k,$$ (2-45)

where $\underline{A}_k^t = \begin{bmatrix} \sigma_k^2 & , & a_1^k & , & \cdots & , & a_p^k \end{bmatrix}$ is the estimated vector of parameters at the k-th iteration, $\underline{G}_k$ is the gradient vector at the k-th iteration and ρ_k is the step size of the algorithm.

Recall that if A_1^k and A_3^k are given, the inverse of the noise covariance matrix is obtained by

$$R_{N,k}^{-1} = \frac{1}{\sigma^2} \left[A_1^k \cdot (A_1^k)^t - A_3^k \cdot (A_3^k)^t \right], \text{ where } A_1^k = \sum_{i=0}^{p} a_i^k Z^i .$$ (2-46)

The problem now is reduced to the computation of the gradient vector $\underline{G}_k$ whose i-th component is given by

$$\underline{G}_k(i) = \frac{\partial}{\partial a_i} L(q,k),\qquad (2\text{-}47)$$

where $L(q,k)$ is the log-likelihood functional conditional to q sources and $R_{N,k}$.

It can be shown that computation of $\underline{G}_k$ amounts to computation of partial derivatives of the eigenvalues of $\hat{R}_{w,k}$. We will first derive some properties in the case of simple eigenvalues and compute the desired partial derivatives.

- First recall that $R_{N,k}$ and $\hat{R}_x$ can be written as $R_{N,k}^{-1} = L_k^{-H} L_k^{-1}$ and $\hat{R} = L_k R_w L_k^H$. Also the identity matrix I can be written as $I_m = L_k^{-H} L_k^H$. Therefore

$$\det\left(R_{N,k}^{-1}\hat{R}_x - \lambda_N I_m\right) = \det\left(L_k^{-H} L_k^{-1} L_k \hat{R}_w L_k^H - \lambda_N L_k^{-H} L_k^H\right) = \det\left[L_k^{-H}\left(\hat{R}_w - \lambda_N I_m\right)L_k^H\right]$$
$$= \det\left(L_k^{-H}\right)\det\left(\hat{R}_{w,k} - \lambda_N I_m\right)\det\left(L_k^H\right) = \det\left(\hat{R}_{w,k} - \lambda_N I_m\right) \qquad (2\text{-}48)$$

This clearly shows that the eigenvalues of $\hat{R}_{w,k}$ and $R_{N,k}^{-1}\hat{R}_x$ are identical.

- Second, note that since $\hat{R}_x$ is a positive matrix, it can be decomposed in triangular factors; i.e., $\hat{R} = TT^H$. Consider then the same determinant as above

$$\det\left(R_{N,k}^{-1}\hat{R}_x - \lambda_N I_m\right) = \det\left(T^{-H} T^H R_{N,k}^{-1} TT^H - \lambda_N T_k^{-H} T_k^H\right) = \det\left[T_k^{-H}\left(R_{N,k}^{-1} - \lambda_N I_m\right)T_k^H\right]$$
$$= \det\left(T_k^{-H}\right)\det\left(R_{N,k}^{-1} - \lambda_N I_m\right)\det\left(T_k^H\right) = \det\left(R_{N,k}^{-1} - \lambda_N I_m\right) \qquad ,(2\text{-}49)$$

which means that the eigenvalues of $\hat{R}_{w,k}$ and $T^H R_{N,k}^{-1} T$ are identical.

From the two properties seen above, it can be deduced that the computation of the partial derivatives $\left(\partial\lambda_j/\partial a_i\right)$ reduces to the computation of the partial derivatives $\left(\partial\mu_j/\partial a_i\right)$, where the μ_j's are the eigenvalues of the Hermitian matrix $T^H R_{N,k}^{-1} T$. This result is very important in the sense that the inverse of the noise covariance matrix is directly related to the AR coefficients.

A classical result in linear algebra states that given a matrix A with its simple eigenvalues λ_j and its corresponding eigenvectors $\underline{V}_j$, the partial derivative $\left(\partial\lambda_j/\partial a_i\right)$ is given by

$$\frac{\partial\lambda_j}{\partial a_i} = \underline{V}_j^H \frac{\partial}{\partial a_i} A \underline{V}_j \text{, where} \left(\frac{\partial}{\partial a_i} A\right)(k,l) = \frac{\partial}{\partial a_i}\left[A(k,l)\right]. \qquad (2\text{-}50)$$

We are now ready to present all the steps involved in the determination of the maximum likelihood functional.

Algorithm

- Computation of partial derivatives Δ_i^k $(0 \le i \le p)$ of N^{-1} relative to the parameters $\{a_i\}$,

$$\Delta_i^k = \frac{\partial}{\partial a_i} R_{N,k}^{-1} = \frac{1}{\sigma^2}\left[Z^i(A_1^k)^t + A_1^k(Z^i)^t - Z^{m-i}(A_3^k)^t - A_3^k(Z^{m-i})^t\right] \text{for } 1 \le i \le p. \quad (2\text{-}51)$$

Note that

$$\Delta_0^k = -2(\sigma_k)^{-1} R_{N,k}^{-1}. \quad (2\text{-}52)$$

- Computation of the derivatives of the matrices $(\Delta_i^k)'$; i.e.,

$$(\Delta_i^k)' = \frac{\partial}{\partial a_i}(T^H R_{N,k}^{-1} T) = T^H \Delta_i^k T ; \ 0 \le i \le p. \quad (2\text{-}53)$$

- Computation of partial derivatives of the (simple) eigenvalues of $\hat{R}_{w,k}$; i.e.,

$$\frac{\partial}{\partial a_i}\lambda_j^k = (\underline{U}_j^k)^H (\Delta_i^k)' (\underline{U}_j^k), \quad (2\text{-}54)$$

where $\underline{U}_j^k$ is the eigenvector associated to the eigenvalue λ_j^k of the matrix $T^H R_{N,k}^{-1} T$.

- Computation of the gradient vector $\underline{G}_k$, defined by its component $G_k(i)$; i.e.,

$$G_k(i) = \sum_{j=q+1}^m (\lambda_j^k)^{-1} \frac{\partial}{\partial a_i}\lambda_j^k - \left[ar^k(q)\right]^{-1} \sum_{j=q+1}^m \frac{\partial}{\partial a_i}\lambda_j^k , \quad (2\text{-}55)$$

where

$$ar^k(q) = \frac{1}{m-q}\sum_{j=q+1}^m \lambda_j^k. \quad (2\text{-}56)$$

In the above algorithm, we have shown all the steps necessary for the computation of the gradient vector $\underline{G}_k$, in the case of a real AR model. However, as noted previously, the algorithm requires the knowledge of the eigensystem of $T^H R_{N,k}^{-1} T$. The eigenvalues and eigenvectors of this matrix can be computed exactly using standard algorithms or they may be

102

estimated as discussed later. It is important to note that this algorithm depends on a computation of a satisfying step size ρ for the gradient algorithm to ensure convergence. This will be achieved by use of a first (or higher)-order approximation to the change in eigenvalues of the whitened matrix. More precisely, let $\underline{\Lambda}_j^k$ be the vector of partial derivatives of λ_j^k; i.e.,

$$(\Lambda_j^k)^t = \left(\frac{\partial}{\partial a_1} \lambda_j^k, \cdots, \frac{\partial}{\partial a_p} \lambda_j^k \right).$$ (2-57)

Then, to a first order approximation, the j-th eigenvalue of the whitened matrix $\hat{R}_{w,k}$ or $\left[R_{N,k}(\rho) \right]^{-1} \hat{R}$ with

$$R_{N,k}^{-1} = \frac{1}{\sigma^2} \left[A_1^k(\rho) \left[A_1^k(\rho) \right]^t - A_3^k(\rho) \left[A_3^k(\rho) \right]^t \right]$$ (2-58)

and

$$A_1^k(\rho) = \sum_{i=0}^{p} \left[a_i - \rho G_k(i) \right] Z^i$$ (2-59)

is given by

$$\lambda_j^k(\rho) \approx \lambda_j^k - \rho \underline{G}_k^t \underline{\Lambda}_j^k.$$ (2-60)

Making these substitutions into the likelihood functional and by means of a uni-dimensional method, an approximation for ρ is determined.

A second order approximation may also be used. In practice, however, a first order approximation seems to be sufficient to ensure convergence of the algorithm. Moreover, it is less computationally extensive.

2.5 - Extensions

In this section, we extend the above analysis to include different parameters and check how the technique performs. First, we extend the technique to include what is referred to as Reflection Coefficients.

103

2.5.1 - Parameterization using reflection coefficients

The parameterization of the matrix R_N may also be expressed in terms of reflection coefficients. In this way, the stability of the AR model is ensured. Specifically, the Levison recursion can be written in matrix form as

$$\underline{A}_{p+1} = \left(I_{p+2} - k_p J\right)\Delta\underline{A}_p,$$ (2-61)

where Δ is the $[(p+2)x(p+1)]$ matrix defined as

$$\Delta = \begin{bmatrix} 1 & \cdots & \underline{0} \\ \underline{0} & \cdots & 1 \\ 0 & \cdots & 0 \end{bmatrix}$$ (2-62)

J is the reflection matrix ($J^2 = I$) and $\underline{A}_p$ is the vector of AR coefficients of the noise model; i.e., $\underline{A}_p^t = [1, a_1, \cdots, a_p]$. It is then possible to express $\underline{A}_{p+1}$ in terms of the reflection coefficients $[1, k_1, \cdots, k_p]$ which leads to

$$\underline{A}_{p+1} = \left(I - k_p J\right)\Delta\left(I - k_{p-1}J\right)\Delta\cdots\left(I - k_1 J\right)\Delta.1.$$ (2-63)

Hence, the partial derivatives of the log-likelihood functional can be expressed with respect to the reflection coefficient as

$$\frac{\partial}{\partial k_i}L_q = \sum_{j=1}^{p}\frac{\partial}{\partial a_j}L_q \cdot \frac{\partial}{\partial k_i}a_j$$ (2-64)

Therefore, the partial derivatives $\left(\partial L_q/\partial k_i\right)$ can easily be obtained since the partial derivatives $\left(\partial L_q/\partial a_j\right)$ have been already derived in the previous section and $\left(\partial a_j/\partial k_i\right)$ can be obtained from Equation (2-63). The test of stability for the polynomial A(z) reduces to the condition

$$0 \le |k_i| \le 1.$$ (2-65)

2.5.2 - ARMA Case

The fact that an MA model is included now leads us to replace the matrix Δ_i^k by the matrix δ_i^k defined by

$$\delta_i^k = -(R_{N,k})^{-1} \cdot \left[\underline{Y}_i \cdot \left(\sum_{l=0}^{p} b_l^k \cdot \underline{Y}_l \right)^t + \left(\sum_{l=0}^{p} b_l^k \cdot \underline{Y}_l \right) \underline{Y}_i^t \right] \cdot (R_{N,k})^{-1}. \qquad (2\text{-}66)$$

The ARMA case is obtained by derivation of Equation (2-43). Note that the MA model is assumed to be minimum phase.

2.5.3 - Multiple Eigenvalues

In the case where some eigenvalues have multiplicity greater than 1, the formula giving the eigenvalues' partial derivatives no longer holds. In this case, up to the first order, the eigenvalues of the perturbed $(R+\delta R)$ are the same as the eigenvalues of the matrix

$$\Delta = U^H \cdot (R + \delta R) \cdot U, \qquad (2\text{-}67)$$

where U is the matrix of an eigenvector basis associated with the multiple eigenvalue. This leads us to replace Equation (2-43) by the computation of the eigenvalues of the matrix $U^H \cdot \left(\Delta_i^k \right)' \cdot U$. The other steps of the algorithm remain the same.

2.5.4 - Complex Case

The fact that the noise field is non symmetric with respect to the array broadside results in complex AR(MA) coefficients for the noise model. However, nothing changes as far as the procedure is concerned.

2.6 - Convergence Analysis

In this section, we study the convergence of the gradient algorithm defined in the previous section. For instance, consider a second order AR model. Its covariance matrix can be computed using Equations (2-37) and (2-38). It is then possible to determine the functional $L_{a_1,a_2}(q)$ as defined in Equation (2-36), where λ_i's are the eigenvalues of the matrix $R_{N,a_1,a_2}^{-1} \cdot R_x$ and a_1 and a_2 are the AR parameters. Note that $L_{a_1,a_2}(q)$ is the exact

log-likelihood functional subject to the condition on $\{a_1, a_2\}$. A 3-dimensional plot of $-L_{a_a,a_2}(q)$ as a function of the AR parameters; i.e. a_1 and a_2, shows that its maximum is attained at the exact values of the AR parameters (see Figures 1-1 and 1-2). However, notice that the function is not concave, however the following properties do apply to the function $L_{a_a,a_2}(q)$:

1. The log-likelihood function is negative. It equals 0 if and only if the noise model is estimated perfectly.

2. The gradient vector of the log-likelihood function is null if and only if A is equal to A_0, which represents the exact values of the coefficients.

In previous sections, we assumed that the number of sources (q) and the AR order (p) are fixed. Next, we will look at the consequences of misadjustment of these parameters.

2.6.1 - Number of sources (q)

Using property 1 above, it is possible to overestimate the number of sources q, as long as the order of the AR model p in less than $(m-q)$ without degradation in the asymptotic case. In practice, the covariance matrix is estimated, therefore the overestimation of q will lead to slightly inferior performances of the method. However, the degradation is quite acceptable. The question, however, is: what would be a good strategy for choosing the value of q. Recall that the classical information criteria such as those developed by Akaike and Rissanen do not provide a satisfying estimation of the number of sources. This is due to the fact that they make use of the eigenvalues of the estimated covariance matrix $\hat{R}_x$ and they do not separate between the uncorrelated sources and (highly) correlated sources. To solve these problems, we present the following solutions

a) **Source over-determination:**

Trying consecutive values of q and then choose the value which maximizes the log-likelihood function will lead us to the right answer.

106

b) Modifications of information criteria:

The information theoretic criteria are based upon log-likelihood determination and its statistical behavior. These statistics are characterized by a number of "free" parameters. The fact that we have an AR noise model enables us to add some "free" parameters. More specifically, assume that Rissanen's criterion is used, in which case, the following function is considered

$$MDL(p,q) = -L(p,q) + \frac{Ln(N)}{2N} \cdot [p(2m-p)+q+2], \qquad (2\text{-}68)$$

where N is the number of independent samples, p is the number of sources and q is the noise model order. L(p,q) is the likelihood function conditional on the following hypotheses: p sources and noise model order q.

c) Statistical properties of the eigenvalues of a "corner" of matrix R (MA case):

This original approach has been presented by Fuchs[12] and gives an accurate estimate of the number of sources in the presence of correlated noise.

d) Use of a State-Space Approach for Sensor Outputs Modeling (ARMA Case):

Using an information criterion such as the predictive efficiency criterion[13], it is possible to obtain satisfactory estimates of the number of sources without a priori knowledge about the noise model.

2.6.2 - Noise Order Model

In the asymptotic case, noise model over-determination leads to very slight degradation of the results in terms of noise spatial density. We will show later, through computer simulations that it is possible to overestimate the order of the noise model, without

---

[12] J. J. Fuchs, "Estimation du Nombre de Sinusoids Dans du Bruit Colore," Actes du Onzieme Colloque GRETSI, Vol. 1, pp. 197-200, June 5, 1987.

[13] K. S. Arun and S. Y. Kung, "Generalized Principal Components analysis and its applications in Approximate Stochastic Realizations," in Modeling and Applications of Stochastic Processes, U. B. Desai, Ed. Boston: Kluwer Academic, 1986, pp. 75-105.

dramatic effects. As shown previously, it is obvious to note that the parameters p and q may be estimated simultaneously. However, a separate estimation of p seems rather difficult.

Another point to consider is the stability of the AR model obtained by the maximizing of the log-likelihood function. It is possible to compute the roots of $\underline{A}_k(z)$, where $\underline{A}_k(z)$ is the polynomial associated with $\underline{A}_k$, and to adjust the step size of the gradient method. However, it can be shown that the parameterization by reflection coefficients is best suited in this case. It is interesting to note that in practice, if the noise poles approach the unit circle, then the noise covariance matrix tends towards singularity. This leads to the log likelihood functional approaching infinity. However, if the step is not very important, this avoids any stability problems.

2.6.3 - Estimation of Source Parameters

Assume that an estimation of the AR parameters has been obtained after several runs of the algorithm described earlier. Let $\underline{A}^t = \left\{ 1 \quad , \quad \hat{a}_1 \quad , \quad \cdots \quad , \quad \hat{a}_p \right\}$ be this estimate.

An application where the above problem arises would be the estimation of Angles Of Arrival (AOA) of some sources as well as their powers at a given frequency. The derivation of source powers estimation is a fundamental tool for the minimization of the whiteness functional. The whitened matrix $\hat{R}_w$ given by $\hat{R}_w = L \cdot R \cdot L^H$, with $R_N^{-1} = L \cdot L^H$, is considered for source bearing estimation. Note that $\hat{R}_w$, is not generally a Toeplitz matrix (even if $\hat{R}_x$ is Toeplitz), however, we can remedy this problem through a filtering process. The steering vectors $\underline{D}_\theta$, corresponding to bearing θ, becomes whitened; i.e., $(L \cdot \underline{D}_\theta)$.

The MUSIC method for AOA estimations consists of computing the sine of the steering vector $(L \cdot \underline{D}_\theta)$ on the source subspace (space spanned by the eigenvectors corresponding to the largest eigenvalues of $\hat{R}_w$,

$$\frac{1}{\sin^2(\theta)} = \frac{\left\| L \cdot D_\theta \right\|^2}{\left\| (I - \Pi) \cdot L \cdot D_\theta \right\|^2} = \frac{\left\| L \cdot D_\theta \right\|^2}{\left\| (\Pi^\perp) \cdot L \cdot D_\theta \right\|^2}, \tag{2-69}$$

where $\| \ \|$ is the Euclidean norm, Π is the projection matrix, $\Pi = V \cdot V^H$ and $V(mxq)$ is the matrix constituted by the greatest eigenvectors of $\hat{R}_w$.

Pisarenko's method has also been considered. Let $\left\{ \hat{\theta}_1 \ , \ \cdots \ , \ \hat{\theta}_q \right\}$ be the estimated angles of the sources. Consider the (mxm) associated matrices; i.e.,

$\left\{ \underline{D}_{\hat{\theta}_i} \cdot \underline{D}_{\hat{\theta}_i}^H \right\}_{i=1}^q$. Then for each of these sources, we pick the sub-matrices

$$S_{\theta_i,l} = \begin{bmatrix} a_{1,1}^i & a_{1,2}^i & \cdots & a_{1,p+1}^i \\ \vdots & \vdots & \cdots & \vdots \\ a_{1+p,1}^i & a_{1+p,2}^i & \cdots & a_{1+p,P+1}^i \end{bmatrix} \tag{2-70}$$

where $a_{l,k}^i$ is the l-th row, k-th column element of the matrix $\left\{ \underline{D}_{\hat{\theta}_i} \cdot \underline{D}_{\hat{\theta}_i}^H \right\}$ for $i = 1, 2, \cdots, s$ and

$l = 1, 2, \cdots, (m-p)$. All the sub-matrices $S_{\hat{\theta},l}$ are Toeplitz, but non-Hermitian. Then the transformed covariances of sources are given by

$$s_w\left(\hat{\theta}_{i,l}\right) = A^H \cdot s\left(\hat{\theta}_{i,l}\right) \cdot A \equiv s_w\left(\hat{\theta}_i, l\right). \tag{2-71}$$

The covariance matrices of the sources $s_w\left(\hat{\theta}_i, l\right)$ correspond to the covariances of the sources after whitening by the inverse filter A as will be seen later. Covariances of transformed array outputs $\hat{r}_w(l)$ are defined by the same method as previously shown, using a Toeplitz estimate of the covariance matrix of the outputs. This estimate is obtained by averaging along the diagonal and is the same as an orthogonal projection on the Toeplitz subspace. Pisarenko's method can then be used for estimating the power of the sources. Assuming spatial whiteness of the additive noise, the estimates of the source powers $\left\{ \hat{g}_1, \cdots, \hat{g}_q \right\}$ are the solutions of the following over-determined linear system

$$\begin{bmatrix} s_w\left(\hat{\theta}_1,1\right) & \cdots & s_w\left(\hat{\theta}_q,1\right) \\ \vdots & \cdots & \vdots \\ s_w\left(\hat{\theta}_1,m-p\right) & \cdots & s_w\left(\hat{\theta}_q,m-p\right) \end{bmatrix} \begin{bmatrix} \hat{g}_1 \\ \vdots \\ \hat{g}_q \end{bmatrix} = \begin{bmatrix} \hat{r}_w(1) \\ \vdots \\ \hat{r}_w(m-p) \end{bmatrix}. \tag{2-72}$$

It is important to observe that the procedure described above is similar to the case of uncorrelated sources which means that the source matrix is given by

$$S = \sum_{i=1}^{q} g_i \underline{D}_{\theta_i} \cdot \underline{D}_{\theta_i}^{H}, \tag{2-73}$$

and theoretically is independent of the noise level since it uses covariances $\hat{r}_w(l)$ with $l \geq 1$.

2.7 - <u>Plane Wave Hypothesis, Whiteness Functional</u>

In this section, we will focus on the case of a linear equispaced array of sensors. We will derive the expressions for the likelihood functional in the case of an AR system, using Pisarenko's technique for harmonic retrieval.

We assume that the covariance matrix R is Toeplitz and that the source parameters are perfectly known and define the function

$$J = \sum_{l=1}^{L} |r(l) - r_s(l)|^2 \; ; (L > 1) \; . \tag{2-74}$$

Note that J is null in the case of white noise. In the case where the noise is correlated, it is possible to use whitening to reach a similar conclusion.

We will only consider the case of an AR model. Let $\underline{A}$ be the vector of coefficients of this model and let $\left[A\left(z^{-1}\right)\right]^{-1}$ be the associated filter. The whitened filter is $A(z^{-1})$ and the covariances of whitened data can be written as

$$r_w(l) = E\left\{\left(\sum_{i=0}^{p} a_i x_{t-i}\right) \cdot \left(\sum_{i=0}^{p} a_i^* x_{t-l-i}^*\right)\right\}, \text{where } a_0 = 1. \tag{2-75}$$

Therefore,

$$r_w(l) = \underline{A}^H \cdot R_1 \cdot \underline{A}, \tag{2-76}$$

where R_1 is the (p+1)x(p+1) matrix given by

$$R_1 = \begin{bmatrix} r(1) & r(1-1) & \cdots & r(1-p) \\ r(1+1) & r(1) & \cdots & r(1-p+1) \\ \vdots & \vdots & \cdots & \vdots \\ r(1+p) & r(1+p-1) & \cdots & r(1) \end{bmatrix}. \tag{2-77}$$

Note that the above matrix is Toeplitz but non-Hermitian. Define the function

$$J_w(\underline{A}) = \sum_{l=1}^{L} |r_w(1) - r_s(1)|^2 , \tag{2-78}$$

where $r_w(1)$ was defined earlier and $r_s(1)$ are the exact covariances of sources after the mapping $\underline{A}$. Then when $\underline{A}$ is equal to $\underline{A}_0$, where $\underline{A}_0$ is the exact model, we can show that

$$r_w(1) = r_s(1) + E\left[\frac{A_0(z^{-1})}{A_0(z^{-1})} \cdot e(t) \cdot \frac{A_0(z^{-1})}{A_0(z^{-1})} \cdot e(t+1) \right] = r_s(1), \tag{2-79}$$

where e(t) is the input white noise sequence for the AR model. Therefore

$$J_w(\underline{A}_0) = 0 . \tag{2-80}$$

For practical applications, one uses the whiteness function defined by

$$J_w(\underline{A}) = \sum_{l=1}^{L} |r_w(1) - \hat{r}_s(1)|^2 , \tag{2-81}$$

where $\hat{r}_w(1)$ was defined earlier, but using the available covariance matrix $\hat{R}_1$ instead of the exact matrix R_1, and $\hat{r}_s(1)$ is estimated using Pisarenko's method or any other high resolution technique applied to the whitened data in the following manner

$$\hat{r}_s(1) = \sum_{i=1}^{m} \hat{q}_i \exp\left\{- j\pi j \hat{f}_i \right\}, \tag{2-82}$$

where m is the number of sensors, $\hat{g}_i$ and $\hat{f}_i$ are the power and spatial frequency of the i-th source, all of which were estimated using a high resolution technique. Note that L is usually chosen to be $L = (m - p - 1)$.

We will now deal with the minimization of the function defined by Equations (2-81) and (2-82). This minimization is done in terms of AR coefficients vector $\underline{A}$ only, but is relative to the terms $\hat{r}_w(1)$ and $\hat{r}_s(1)$. The general procedure is summarized below:

- Iteration 0, (Starting Parameters):

$$\underline{A}_0^t = [1, 0, \cdots, 0]$$

$\hat{g}_i^0$ and $\hat{f}_i^0$ obtained by spatial analysis of $\hat{R}$

$$\hat{r}_w(1) = \hat{r}(1)$$

- Iteration k:

$$\underline{A}_k^t = \left[1, a_1^k, \cdots, a_p^k\right]$$

$\hat{g}_i^k$ and $\hat{f}_i^k$ obtained by spatial analysis of $\hat{R}_w^k$

$$\hat{r}_w^k(1) = \underline{A}^H \cdot \hat{R}_1 \cdot \underline{A}.$$

Note that the number of sources may be corrected at each iteration.

- Iteration k→ (k+1): Computation of the gradient vector $\underline{G}_J(\underline{A}_k)$

$$\underline{A}_{k+1} = \underline{A}_k - \rho_k \cdot \underline{G}_J(\underline{A}_k), \text{ where } \rho_k \text{ is the gradient step size}.$$

2.7.1 - Whitening Invariance Properties

Recall that $r_w(1)$ can be expressed in matrix form as $R_w = A \cdot R_x \cdot A^H$, where

$$A = \begin{bmatrix} 1 & a_1 & \cdots & a_p & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & & & & & & & & & \underline{0} \\ \vdots & & \underline{0} & & & & & & & \\ 1 & \cdots & 0 & 1 & a_1 & \cdots & a_p & 0 & \cdots & 0 \end{bmatrix}. \tag{2-83}$$

Here A is the directional matrix and therefore should not be confused with the gradient vector discussed in section 2.7. Note also that the matrix R_w defined earlier is a Toeplitz positive

matrix. Moreover, we have $R_w(1,1) = r_w(1)$. This matrix transformation allows us to study the effect of whitening on the covariance matrix.

Next, we will study the effect of whitening on a plane wave whose associated steering vector is given by

$$\underline{D}_\theta^t = \begin{bmatrix} 1 & , & e^{j\alpha} & , & \cdots & , & e^{j(n_s-1)\alpha} \end{bmatrix}. \tag{2-84}$$

It can then easily be seen that the covariance matrix R_w is given by

$$R_w(\theta) = A \cdot \underline{D}_\theta \cdot \underline{D}_\theta^H \cdot A^H. \tag{2-85}$$

Note that the product $A\underline{D}_\theta$ is given by

$$A\underline{D}_\theta = \begin{bmatrix} 1 + a_1 e^{j\alpha} + \ldots + a_p e^{j(p-1)\alpha} \\ e^{j\alpha} + a_1 e^{2j\alpha} + \ldots + a_p e^{jp\alpha} \\ \vdots \\ e^{j(L-1)\alpha} + a_1 e^{j(L)\alpha} + \ldots + a_p e^{j(L+p-1)\alpha} \end{bmatrix}, \tag{2-86}$$

which can also be written as

$$A\underline{D}_\theta = \left(1 + a_1 e^{j\alpha} + \ldots + a_p e^{j(p-1)\alpha} \right) \underline{D}_\theta^{'H}, \tag{2-87}$$

where $\underline{D}_\theta^{'}$ is and (Lx1) vector obtained by selecting the first L components of the vector $\underline{D}_\theta$. Hence, it can easily be shown that the covariance matrix can be expressed as

$$R_w(\theta) = q(\theta) \cdot \underline{D}_\theta \cdot \underline{D}_\theta^H, \tag{2-88}$$

where

$$q(\theta) = \left| 1 + a_1 e^{j\alpha} + \ldots + a_p e^{j(p-1)\alpha} \right|^2. \tag{2-89}$$

The above properties lead to the following:

- Let R_x be a Toeplitz covariance matrix, then $R_w = A \cdot R_x \cdot A^H$ is a Toeplitz matrix.

- The procedure transforms a steering vector into another steering vector corresponding to the same angle of arrival.

In practice, the exact angle of arrival of the sources are unknown. The above properties state that only these angles are invariant under the whitening procedure. These angles may be estimated using any high resolution technique. It is well known, however, that these estimates will never be exact. Using the whiteness function, the intent is to define an iterative method which will minimize the differences between the off-diagonal terms of the whitened matrix and the corresponding terms of the source matrix.

2.7.2 - <u>Calculation of the Gradient Vector of the Whiteness function</u>

Using Equations (2-81) and (2-83), we can show that the gradient vector is given by

$$\underline{G}_J(\underline{A}) = 2\operatorname{Re}\left\{ \sum_{i=1}^{L} [R_1 \underline{A} - \underline{Gr}_s(1)][\underline{A}^H \cdot R_1^H \cdot \underline{A} - r_s^*(1)] \right\}, \qquad (2\text{-}90)$$

where $\underline{Gr}_s(1)$ is the gradient vector of $r_s(1, \underline{A})$ relative to $\underline{A}$. The hardest part of the problem is to determine this last quantity. Note that $\underline{Gr}_s(1)$ is defined by

$$\underline{Gr}_s(1) = \sum_{i=1}^{n} \underline{G}\hat{g}_i(\underline{A}) \cdot e^{-j\pi l \hat{f}_i} - j\pi l \cdot \sum_{i=1}^{n} \hat{g}_i e^{-j\pi l \hat{f}_i} \cdot \underline{G}\hat{f}_i(\underline{A}). \qquad (2\text{-}91)$$

Note that the computation of $\underline{Gr}_s(1)$ amounts to the computation of $\underline{G}\hat{f}_i(\underline{A})$ and $\underline{G}\hat{g}_i(\underline{A})$. This will be achieved in two steps. First, we compute the partial derivatives of the eigenvectors of R_w. Then, derivatives of the source parameters are computed using a perturbation analysis of a high resolution technique.

1. **Partial Derivatives of Eigenvectors of** R_w

Denote by $\hat{R}_A$ the (L+1)x(L+1) Toeplitz matrix defined by

$$\hat{R}_A(i-j) = \hat{r}_A(i-j), \qquad (2\text{-}92)$$

with $\hat{r}_A(1) = \underline{A}^H \cdot R_1 \cdot \underline{A}$ and $\{\underline{U}_1, \cdots, \underline{U}_{L+1}; \lambda_1 \geq \cdots \geq \lambda_{L+1}\}$ is the eigensystem of $\hat{R}_A$. In the case of a simple eigenvalue, it can be shown that

114

$$\frac{\partial}{\partial a_i} \underline{U}_k = \sum_{\substack{j=1 \\ j \neq k}}^{L+1} \left\{ \frac{\underline{U}_j^H \cdot \left(\partial \hat{R}_A / \partial a_j \right) \cdot \underline{U}_k}{\lambda_k - \lambda_j} \right\} \cdot \underline{U}_j \; ; \; (n+1 \leq k \leq L+1). \qquad (2\text{-}93)$$

The matrix $\left(\partial \hat{R}_A / \partial a_j \right)$ is a Toeplitz matrix which can be defined by its first row as

$$\frac{\partial}{\partial a_j} \hat{R}_A (m,1) = \underline{E}_i^H \cdot R_{m-1} \cdot \underline{A} + \underline{A}^H \cdot R_{m-1} \cdot \underline{E}_i , \qquad (2\text{-}94)$$

with

$$\underline{E}_i^t (1) = \begin{cases} 1; 1 = i \\ 0; \text{else} \end{cases} . \qquad (2\text{-}95)$$

2. Derivatives of Source Bearings

Recall that the i-th source is characterized by its power and spatial frequency $\hat{g}_i$ and $\hat{f}_i$, respectively. The problem now consists on computing their partial derivatives with respect to the parameters $\{a_i\}$. In the case where the MUSIC algorithm is used, recall that the estimates of the angles of arrival are obtained by minimizing the projection of vector $\underline{D}_\theta'$ on the noise subspace: $\Pi = U \cdot U^H$, where $U = [\underline{U}_{n+1} , \cdots , \underline{U}_L]$ ("lowest" eigenvectors of R_w).

The projection of $\underline{D}_\theta'$ on the noise sub-space is given by:

$$\Pi(\theta) = \underline{D}_\theta'^H \cdot \Pi \cdot \underline{D}_\theta' . \qquad (2\text{-}96)$$

Let $\underline{\hat{Z}}_i^t = \begin{bmatrix} 1 & , & \hat{z}_i & , & \cdots & , & \hat{z}_i^L \end{bmatrix}$ with $\hat{z}_i = \exp \left\{ -j\pi \hat{f}_i \right\}$. Then the derivative of this vector with respect to the spatial frequencies is given by

$$\frac{\partial}{\partial \hat{f}_i} \underline{\hat{Z}}_i^t = -j\pi \begin{bmatrix} 0 & , & \hat{z}_i & , & \cdots & , & L\hat{z}_i^L \end{bmatrix} . \qquad (2\text{-}97)$$

Then an expression of the partial derivatives of the spatial frequencies with respect to the AR coefficients is expressed as

$$\frac{\partial}{\partial a_1}\hat{\underline{Z}}_i^H \cdot \Pi \cdot \hat{\underline{Z}}_i + \hat{\underline{Z}}_i^H \cdot \Pi \cdot \frac{\partial}{\partial a_1}\hat{\underline{Z}}_i + \hat{\underline{Z}}_i^H \cdot \frac{\partial}{\partial a_1} \cdot \hat{\underline{Z}}_i = 0$$

So far we have shown how to compute the partial derivatives of the eigenvectors using equation (2-93), the spatial frequencies $\hat{f}_i$ using a high resolution technique and the partial derivatives $\left(\partial\hat{f}_i / \partial a_1\right)$. If the projection of the whole noise subspace is considered, the derivative of the projector Π is given by

$$\frac{\partial}{\partial a_1}\hat{f}_i = -\frac{1}{\pi} \cdot \frac{Re\left\{\left[\left(\frac{\partial}{\partial a_1}\underline{U}\right)^H \cdot \hat{\underline{Z}}_i\right] \cdot \left[\hat{\underline{Z}}_i^* \cdot \underline{U}\right]\right\}}{Im\left\{\left(\underline{U}^H \cdot \hat{\underline{Z}}_i'\right) \cdot \left(\hat{\underline{Z}}_i^* \cdot \underline{U}\right)\right\}}. \tag{2-98}$$

and

$$\frac{\partial}{\partial a_1}\Pi(\theta) = \left(\frac{\partial}{\partial a_1}\underline{D}_\theta'\right)^H \cdot \Pi \cdot \underline{D}_\theta' + \underline{D}_\theta'^H \cdot \left(\frac{\partial}{\partial a_1}\Pi\right) \cdot \underline{D}_\theta' + \underline{D}_\theta'^H \cdot \Pi \cdot \left(\frac{\partial}{\partial a_1}\underline{D}_\theta'\right). \tag{2-99}$$

The next step will involve computing the partial derivatives of the projector Π. To do this, we take a second order of Π as given by

$$\Pi_\delta = \left(U_0 + \delta U + \delta^2 U\right) \cdot \left(I - \|\delta U\|^2\right) \cdot \left(U_0^H + \delta U^H + \delta^2 U^H\right), \tag{2-100}$$

where

$$\underline{U}_i \equiv \left(\underline{U}_i\right)_0 + \left(\delta\underline{U}_i\right) + \left(\delta^2\underline{U}_i\right) \tag{2-101}$$

and

$$\left(U_0 + \delta U + \delta^2 U\right)^H \cdot \left(U_0 + \delta U + \delta^2 U\right) = I + \|\delta U\|^2. \tag{2-102}$$

From Equations (2-100), (2-101) and (2-102), we can get a first order expansion of Π_δ as

$$\Pi_\delta = \Pi + U \cdot \delta U^H + \delta U \cdot U^H. \tag{2-103}$$

Therefore

116

$$\frac{\partial}{\partial a_l} \Pi = U \cdot \left(\frac{\partial}{\partial a_l} U \right)^H + \left(\frac{\partial}{\partial a_l} U \right) \cdot U^H, \tag{2-104}$$

from which we obtain

$$\frac{\partial}{\partial a_l} \hat{f}_i = \frac{-1}{(L-n)\pi} \cdot \sum_{k=n+1}^{L} \frac{\mathrm{Re}\left\{ \left[\left(\frac{\partial}{\partial a_l} \underline{U}_k \right)^H \cdot \hat{\underline{Z}}_i \right] \cdot \left[\hat{\underline{Z}}_i^H \cdot \underline{U}_k \right] \right\}}{\mathrm{Im}\left\{ \left(\underline{U}_k^H \cdot \hat{\underline{Z}}_i \right) \cdot \left(\hat{\underline{Z}}_i^H \cdot \underline{U}_k \right) \right\}}. \tag{2-105}$$

It was possible to have used a second order expansion. However, in most practical situations, a first order expansion yields good results.

3. Derivatives of Source Powers

Let F be an (Lxq) matrix, obtained from elementary theoretical covariances of estimated source; i.e.,

$$F = \begin{bmatrix} s(\hat{\theta}_1, 1) & \cdots & s(\hat{\theta}_q, 1) \\ \vdots & \cdots & \vdots \\ s(\hat{\theta}_1, L) & \cdots & s(\hat{\theta}_q, L) \end{bmatrix} \text{ with } s(\hat{\theta}_k, 1) = -\sin(\pi l \hat{f}_k). \tag{2-106}$$

Moreover, let $\underline{J}$ be the (Lxq) vector obtained from the imaginary parts of whitened output covariances; i.e.,

$$\underline{J}^t = \mathrm{Im}\left[\hat{r}_A(1), \cdots, \hat{r}_A(L) \right], \tag{2-107}$$

where $\hat{r}_A(l)$ and A were defined previously. Also, let $\underline{\Gamma}$ be the vector of source powers defined as

$$\underline{\Gamma}^t = \begin{bmatrix} g_1 & , & \cdots & , & g_q \end{bmatrix}. \tag{2-108}$$

In the white noise case, the vector $\underline{\Gamma}$ is the solution to the following linear system

$$F \cdot \underline{\Gamma} = \underline{J}. \tag{2-109}$$

The gradient vector $\underline{G}_l$ where

117

$$\underline{G}_1^t = \left[\frac{\partial q_1}{\partial a_1} \quad , \quad \cdots \quad , \quad \frac{\partial q_s}{\partial a_1} \right],$$ (2-110)

is obtained by differentiating Equation (2-109) and is given by

$$\underline{G}_1 = -\left(F^{-1} \cdot \frac{\partial F}{\partial a_1} \cdot F^{-1} \right) \cdot \underline{J} + F^{-1} \cdot \left(\frac{\partial \underline{J}}{\partial a_1} \right),$$ (2-111)

where

$\frac{\partial}{\partial a_1} F$ is obtained from Equations (2-93) and (2-105), $\frac{\partial}{\partial a_1} \underline{J}$ is given by Equations (2-94) and (2-95) and F^{-1} is the pseudo-inverse matrix of F. The vectors $\underline{G}_{\hat{q}_i}(A)$ are obtained from the vectors $\underline{G}_1$.

3

Computer Simulations and Results

In this section, we present the results of the computer simulations which we carried out. The goal of the simulations is to demonstrate the effectiveness of the proposed techniques. Note that at this stage only the Parametric beamforming algorithm based on LeCadre's technique was evaluated. Note also that only spatial processing was considered.

Recall that the objective of the proposed method is to determine the number of targets present in the environment. Therefore the detection problem which we undertook in this effort is different from the classical detection problem which we usually refer to in the literature as a decision making about the presence or absence of a target. LeCadre's technique is believed to improve the detection of targets, especially for cases of low signal-to-noise ratios. This is of great interest to us since in most applications the targets are buried in the environment consisting of clutter, jammers and thermal noise. Also, by being parametric, the technique shows great promise in the overall detection scheme such as computational efficiency.

3.1 - Likelihood Functional

The first goal of the computer simulations was to duplicate the results presented in [1], and then extend the technique to more complicated scenarios. Preliminary computer simulated results show that the technique performs as expected, since we were able to duplicate most of the results presented by J. P. LeCadre. For instance, we showed that the likelihood functional defined in the paper is minimized (or maximized, depending on which quantity we are using) with respect to the Auto Regressive (AR) coefficients of the noise model and this was done for a fixed and known number of sources. The scenario used in this case was as follows. It consisted of three (q = 3) targets with parameters as shown in the table below:

Target	θ	SNR (dB)
1	40	-5
2	50	-10
3	70	-15

Table 3-1. <u>Target Parameters</u>

The AR model was assumed to be of order two (p = 2) with the following AR parameters:

$$a_0 = 1, \quad a_1 = -0.9, \quad a_2 = 0.2 .$$

We considered a linear uniformly spaced array consisting of 14 sensors (m = 14), spaced at half wavelength $d = \lambda / 2$, such that $\omega d / c = 2\pi (d/\lambda) = \pi$. 500 samples (snapshots) were used in the computation of the sample covariance matrix. The value of the likelihood functional was found to be L = 0.1582. We varied the values of a_1 from -2 to +2 and the value of a_2 from -1 to +1. Note that throughout the analysis as well as the computer simulations, the value of a_0 is assumed to be equal to unity (a_0 = 1). Figures 3-1 and 3-2 show the variations of the likelihood functional as a function of a_1 and a_2.
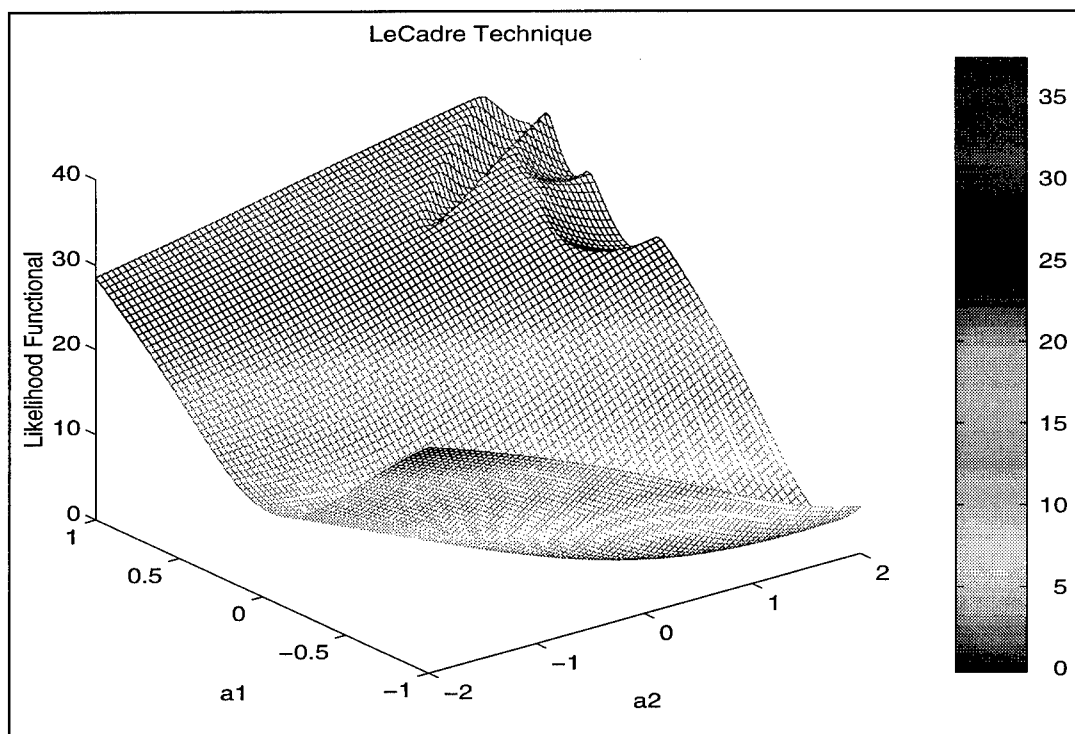
Figure 3-1. Likelihood Functional as a Function of a_1 and a_2
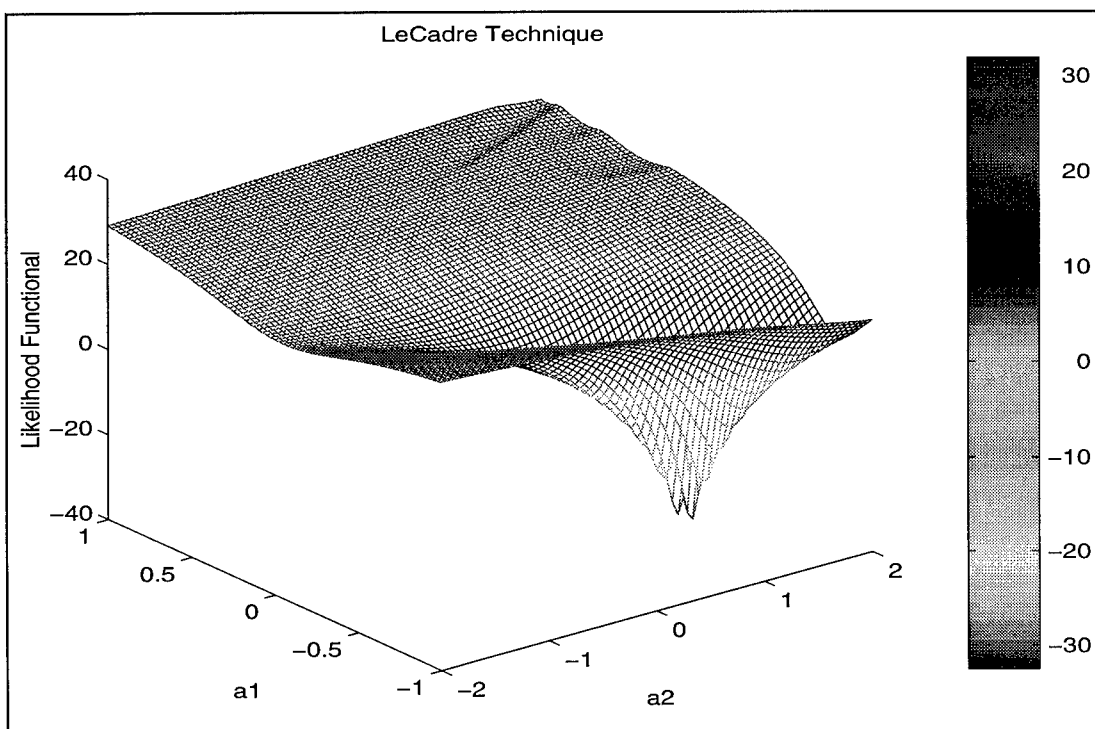
Figure 3-2. Likelihood Functional (in dB) as a Function of a_1 and a_2

121

From the above figures, it can be seen that the minimum of the likelihood functional is attained at the exact values of the AR parameters, namely a_1 and a_2. This shows that the algorithm performs exactly as expected.

3.2 - AR Parameters Estimation

Another aspect of the technique is to accurately estimate the order as well as the coefficients of the AR model used to characterize the noise. In the next simulation, the same scenario as previously described is used. We also assume that the order of the AR model and the number of sources have accurately been estimated. The following figures show the AR coefficients estimates as a function of the step size (ρ) used and the number of iterations needed for convergence. Note that in all cases, we have assumed a total number of 50 iterations. However, as can be seen from these figures, convergence was attained much faster than 50 iterations depending on the value of the step size. In the figures below, the value of ρ was varied from 0.4 to 0.7 with increments of 0.05. As can be seen from these figures, the bias errors seem to have a sinusoidal relation to the step size.
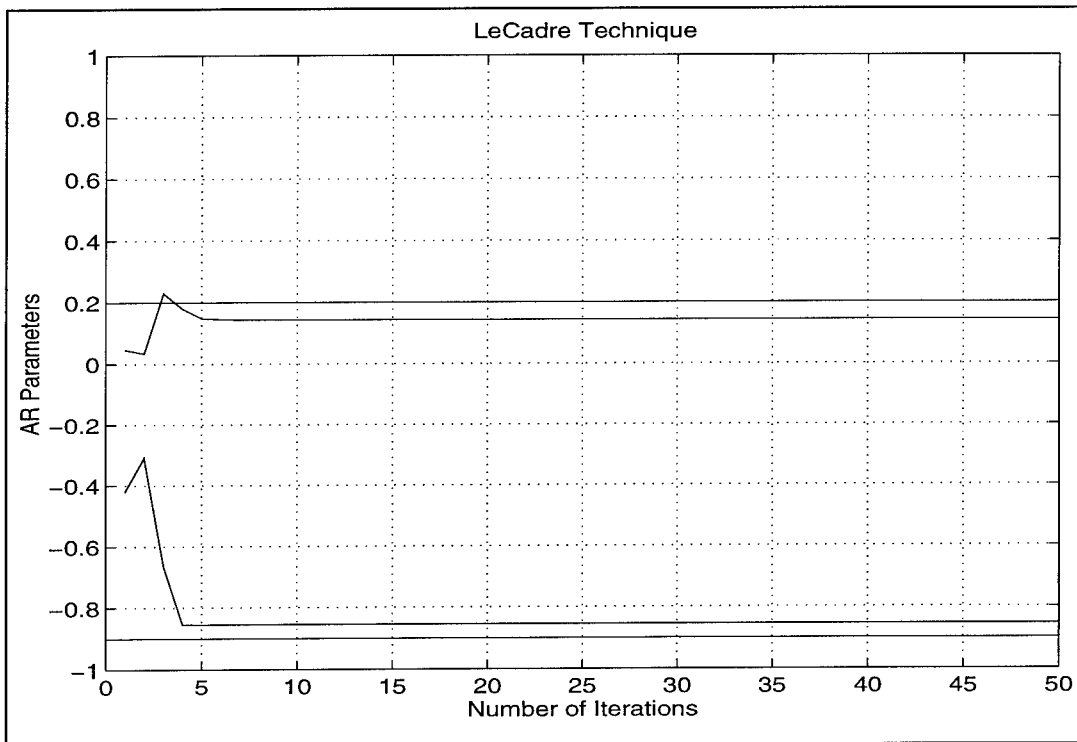
Figure 3-3. <u>AR Parameter Estimates (step size = 0.4)</u>

122
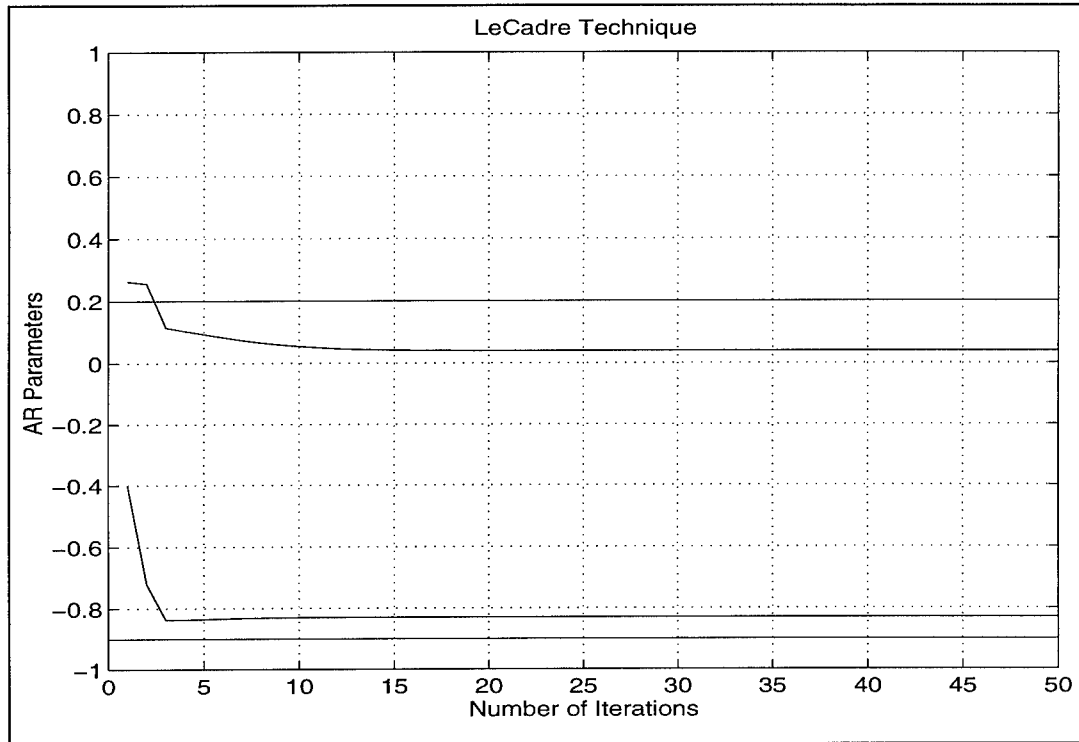
Figure 3-4. <u>AR Parameter Estimates (step size = 0.45)</u>
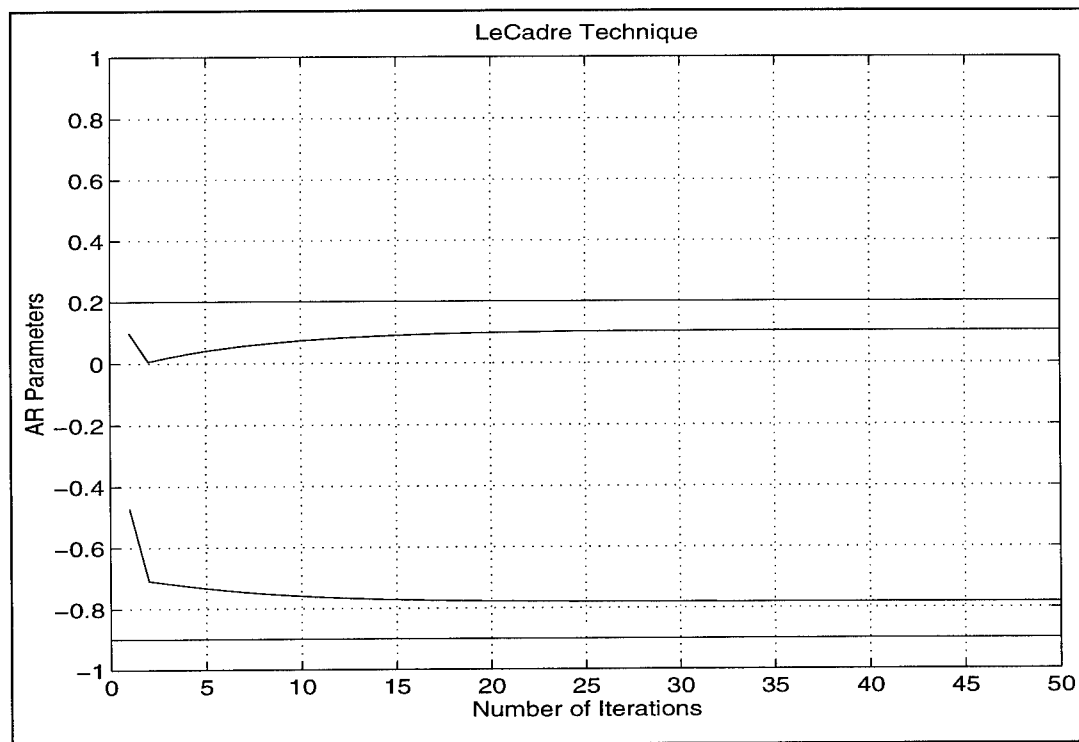
Figure 3-5. <u>AR Parameter Estimates (step size = 0.5)</u>

123
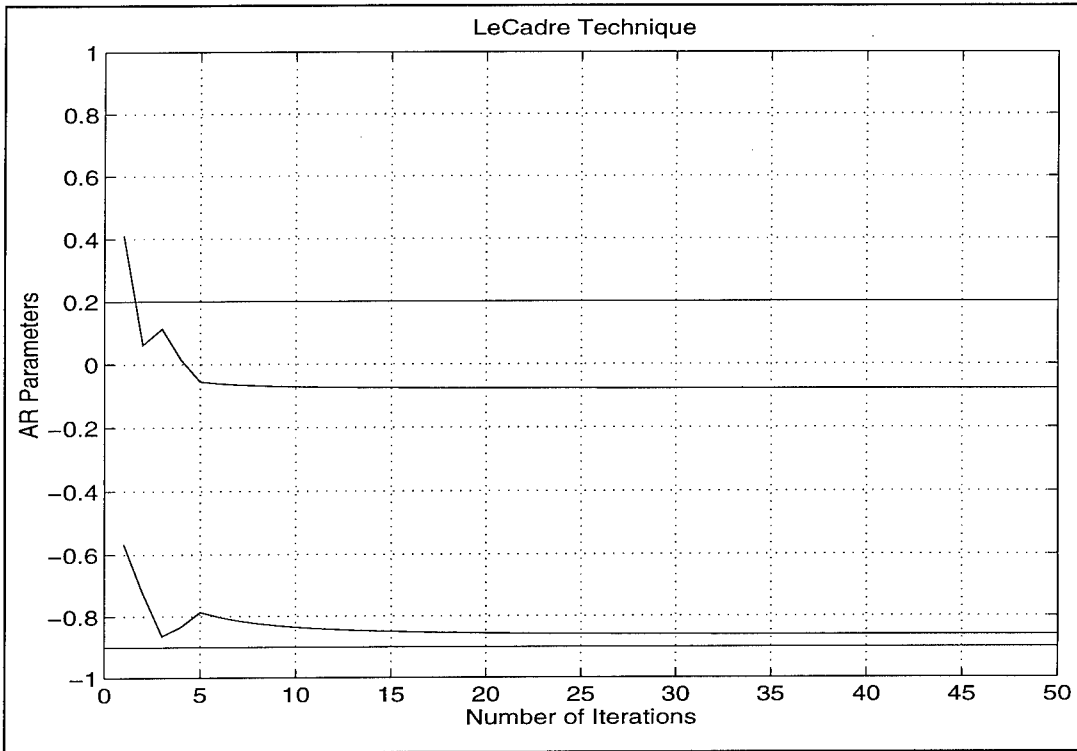
Figure 3-6. <u>AR Parameter Estimates (step size = 0.55)</u>
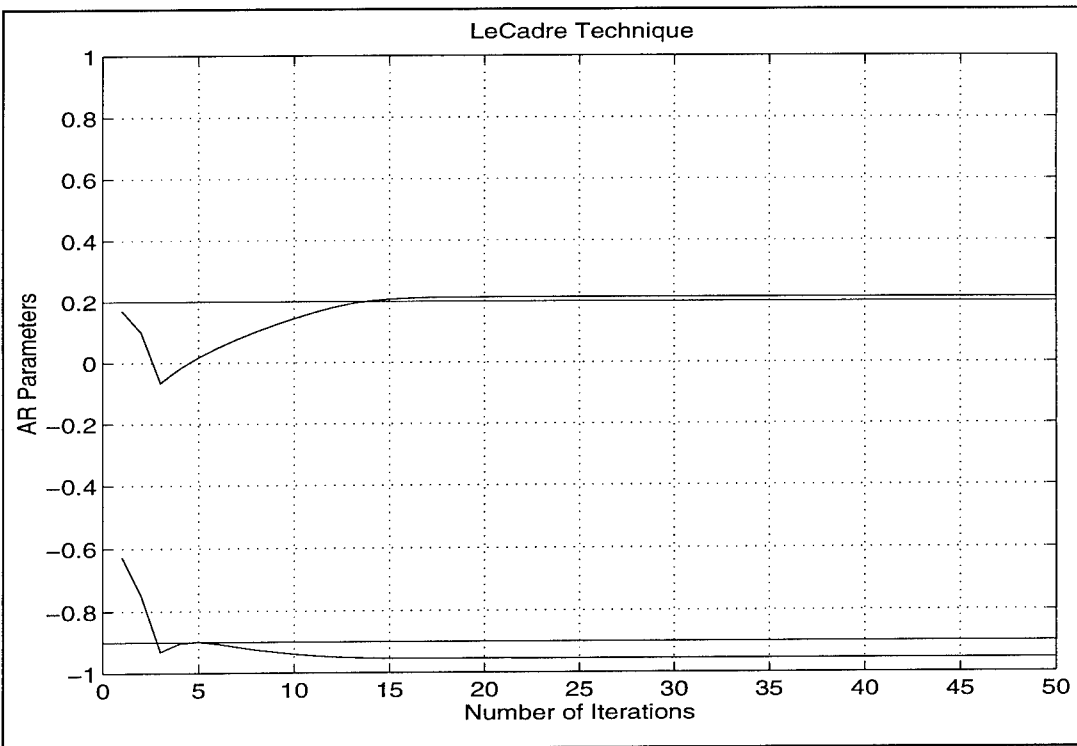
Figure 3-7. <u>AR Parameter Estimates (step size = 0.6)</u>

124
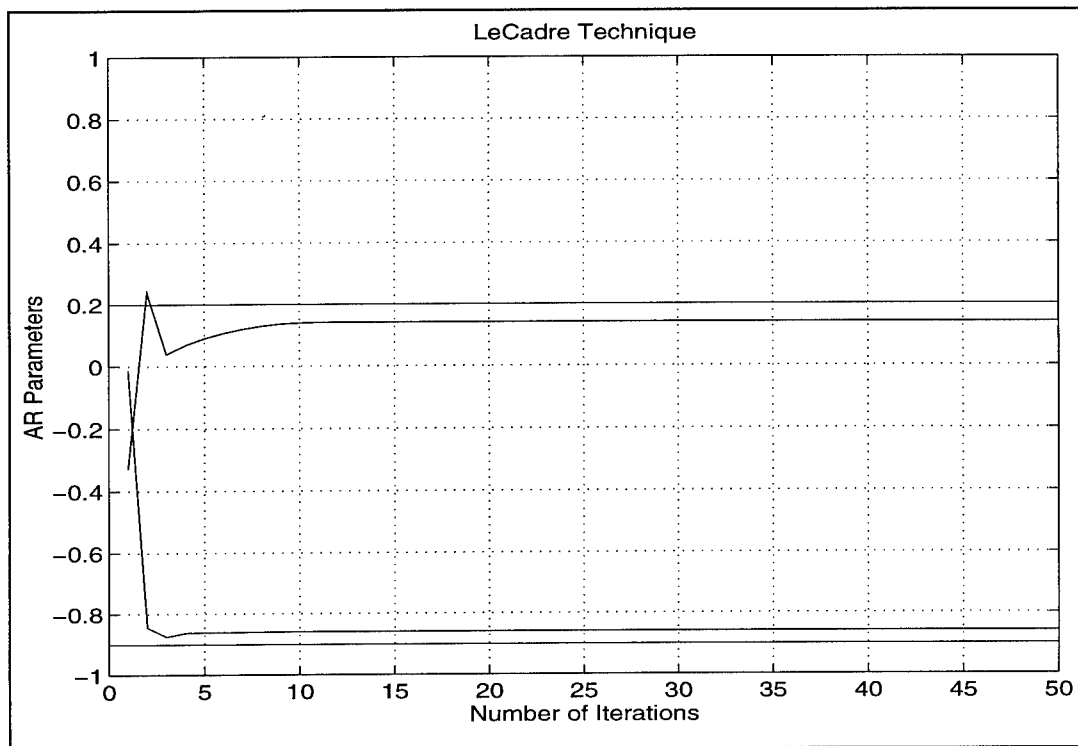
Figure 3-8. <u>AR Parameters Estimates ($\rho = 0.65$)</u>
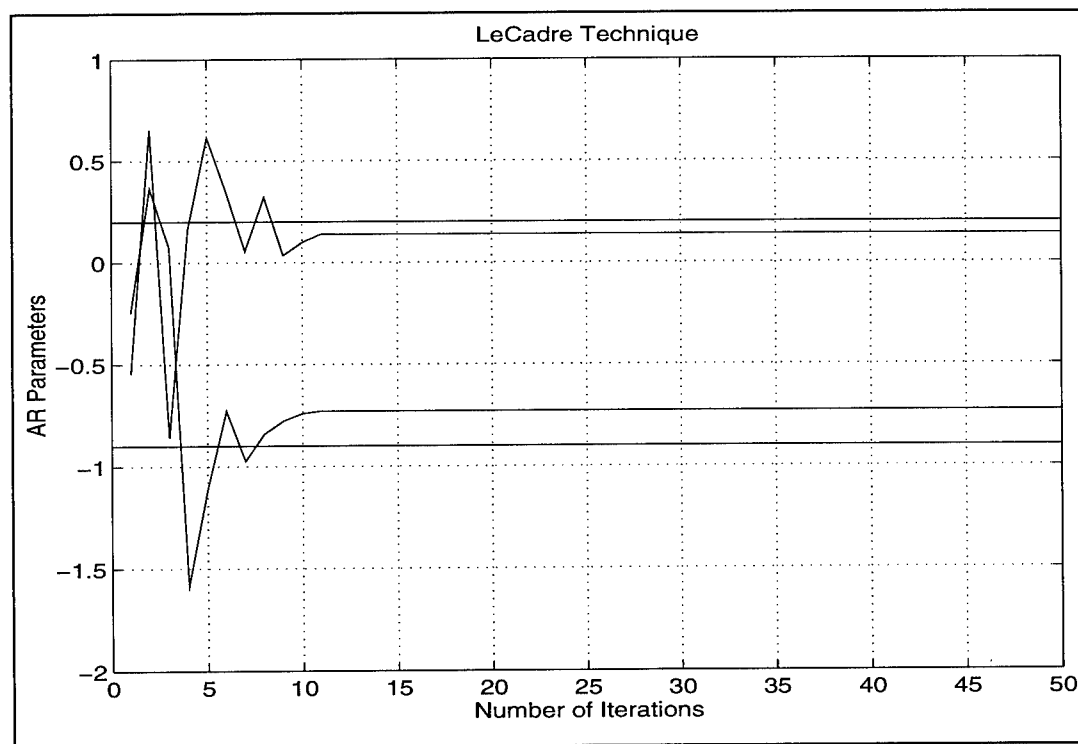
Figure 3-9. <u>AR Parameter Estimates ($\rho = 0.7$)</u>

125

From Figures (3-3) to (3-9), the following observations were made. With a step size of 0.4, convergence was attained after 5 iterations (very fast). However, there is a bias of about 0.02 for both a_1 and a_2. This is not bad, depending on whether the technique is robust to errors in the AR parameters or not. This will discussed later. Note that with a step size of 0.45, convergence was attained after 15 iterations, however with still a bias of about 0.18 for a_1 and 0.05 for a_2. With a step size of 0.5, convergence was attained after 20 iterations, however again with a bias of approximately 0.05 for both a_1 and a_2. With a step size of 0.55, convergence was attained after 15 iterations for both a_1 and a_2, with a very large bias of about 0.25 for a_1 and a small bias of about 0.02 for a_2. With a step size of 0.6, convergence was attained after 15 iterations, however with a very small bias of approximately 0.0005 for a_1 and a bias of about 0.02 for a_2. With as step size of 0.65, convergence was attained after 5 iterations for a_2 with a bias of 0.002 and after 10 iterations for a_1 with a bias of approximately 0.003. When the value of the step size was 0.7, convergence was attained after 12 iterations for both a_1 and a_2, however with bias of about 0.0015 for a_1 and about 0.25 for a_2. From the above discussion, it is clear that the best results were obtained when ρ was equal to 0.6. This value should be the one to be used in further processing of the received signals.

3.3 - <u>Number of Signal Estimation</u>

Next, we show that an extension of the MDL algorithm can also be used to determine the number of sources and the order of the AR model. Computer simulations have shown that the minimum of the functional was attained at the exact values of these parameters. Again, as in the previous case, we considered the same scenario as before. We applied the modified MDL criterion as defined earlier. Note that these results were achieved through whitening of the covariance matrix of the received vector. Table 2 shows the results of the computer simulation.

Process Order	0	1	2	3	4	5
Number of Signals						
1	45.2716	29.4954	3.0568	3.4702	3.8657	4.2586
2	45.2431	29.4668	3.0283	3.4417	3.8372	4.2301
3	45.2336	29.4573	3.0188	3.4322	3.8277	4.2206
4	45.3001	29.5239	3.0853	3.4987	3.8942	4.2872
5	45.4997	29.7235	3.2849	3.6983	4.0939	4.4868
6	46.5264	30.0113	3.6747	4.0881	4.4836	4.8765

Table 3-1. **Values of New MDL Criterion**

As expected, the minimum value of the new MDL criterion was obtained when we reached the exact value of the AR order model and the exact number of sources. From this table we deduce both the order of the AR model (p = 2) and the number of sources (q = 3) for subsequent analysis. It is important to note here the efficacy of the algorithm in estimating these two important parameters. Without the knowledge of these two parameters, it would be extremely hard to process the data any further, especially if we are interested in estimating the locations of the sources.

3.4 - Angle of Arrival Estimation

With the above knowledge in hand, we applied two different algorithms to locate the targets involved in the scenario, namely MUSIC[14] and ESPRIT[15] . It is important to note here that two different simulations were carried out.

3.4.1 - Signal Plus Noise Case With Known Whitening Functional

First, we assumed that the received signal was of the following form:

$$\underline{Y} = A\underline{S} + C\underline{N},$$

---

[14] R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," IEEE Trans. Antennas and Propagation, Vol. AP-34, pp. 276-280, March 1986.

[15] A. Paulraj, R. Roy and T. Kailath, "Estimation of signal parameters via rotation invariance technique-ESPRIT," Proc. 19-th Asilomar Conf. on Circuits, Systems and Computers, pp. 83-89, Pacific Grove, CA, Nov. 1985.

where A is the directional matrix (representing the array manifold), $\underline{S}$ is the impinging signal vector and C is the noise driving term and $\underline{N}$ is the additive white noise vector assumed to be independent from the signals. As described earlier, C was obtained from the Cholesky decomposition of the AR covariance matrix and thus provides the specified cross-channel correlation. We refer to this matrix as the whitening functional. At first, we assumed this matrix to be known a priori, which means that the AR covariance matrix was computed exactly knowing the exact values of the AR parameters.

3.4.1.1 - MUSIC Operator

Figure 3.10 shows the result of the MUSIC algorithm. The solid line shows the transformed MUSIC algorithm, while the dotted line shows the un-whitened MUSIC algorithm. It is very clear from the above figure that whitening does improve the location of the signals. Without any whitening, one sees that MUSIC was unable to resolve all targets. Recall that the signals are completely buried in the clutter (SNR of -5, -10 and -15 dB). These results clearly show the need for the whitening functional.
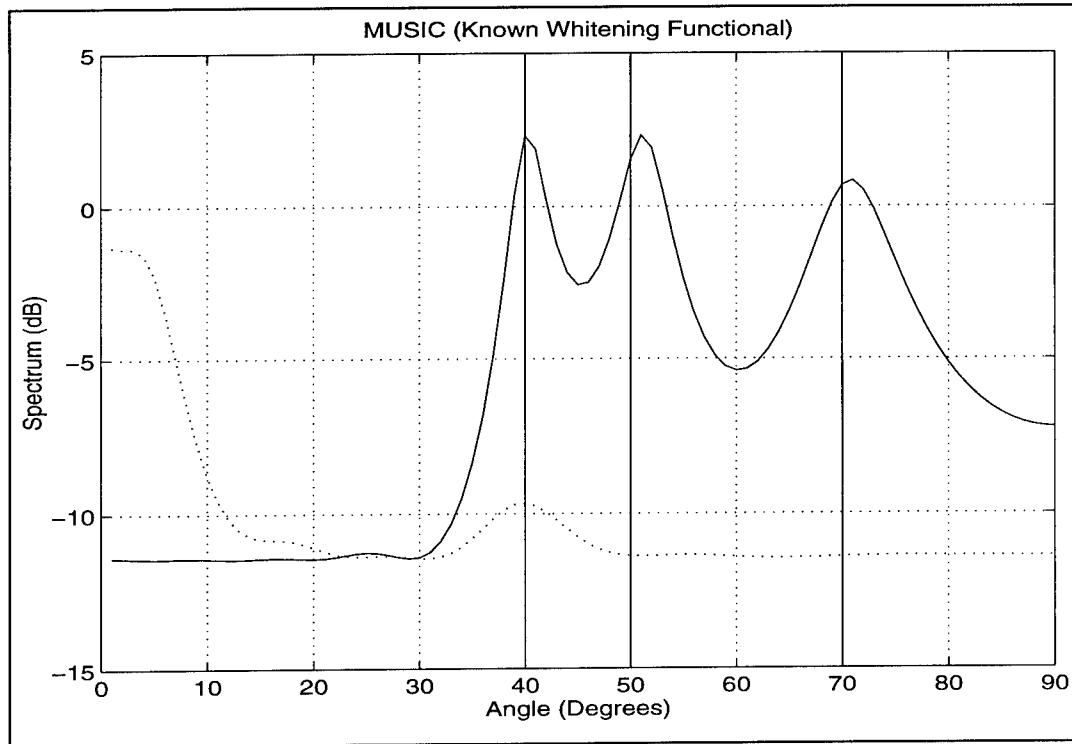
Figure 3-10. MUSIC Algorithm

128

3.4.1.2 - <u>ESPRIT Operator</u>

In the previous section, we concentrated on using LeCadre's technique in conjunction with the MUSIC algorithm. It is well known that one of the disadvantages of MUSIC is the computational load required to execute the algorithm as well as the amount of memory needed to store the array manifold. For these reasons, we shifted our attention to another technique which is computationally less intensive and does not require a lot of memory space. This technique is referred to as ESPRIT (Estimation of Signal Parameters via a Rotational Invariance Transformation). In this technique, two sub-arrays are formed. Then, we form the covariance matrix of the first sub-array, call it M_1, and the cross-covariance matrix of sub-array 1 and sub-array 2, call it M_2. It can then be easily shown that the information of the locations of the targets is contained in the rank reducing values of the matrix pencil $(M_1 - \lambda M_2)$. These rank reducing values are shown to be the generalized eigenvalues of the same matrix pencil. In the initial phase, we assumed that the whitening transformation is known a priori. We have selected the same scenario as described previously in which we considered an array consisting of 14 sensors spaced at half wavelength and 3 targets located at 40°, 50° and 70°, with an SNR of 0 dB, each. The AR process was still assumed to be of order 2 with parameters 1, -0.9 and 0.2. The following show the mean, the variance and the mean-squared error of the estimates obtained after 50 trials.

	$\theta_1 = 40°$	$\theta_2 = 50°$	$\theta_3 = 70°$
Mean	41.0609	51.3136	70.6455
Variance	0.0036	0.0030	0.0057
MSE	56.2737	86.2731	20.8348

Table 3-3. <u>ESPRIT Estimates</u>

We can also visualize these results by looking at the angle estimates as a function of the number of iterations and the histograms of the estimates.
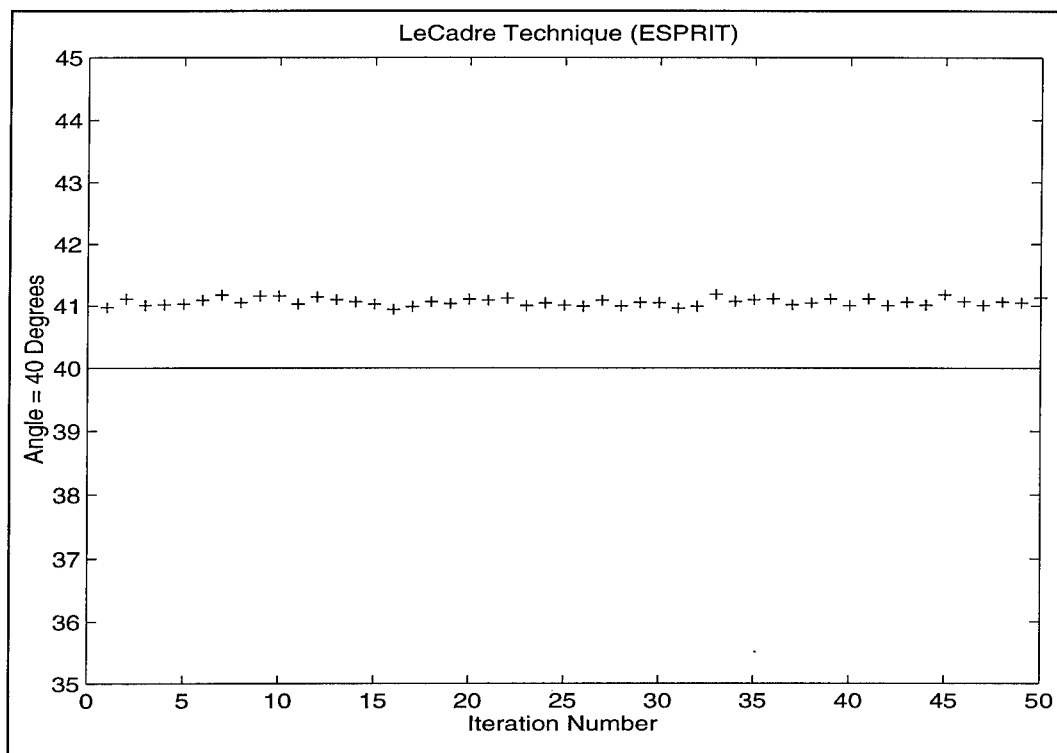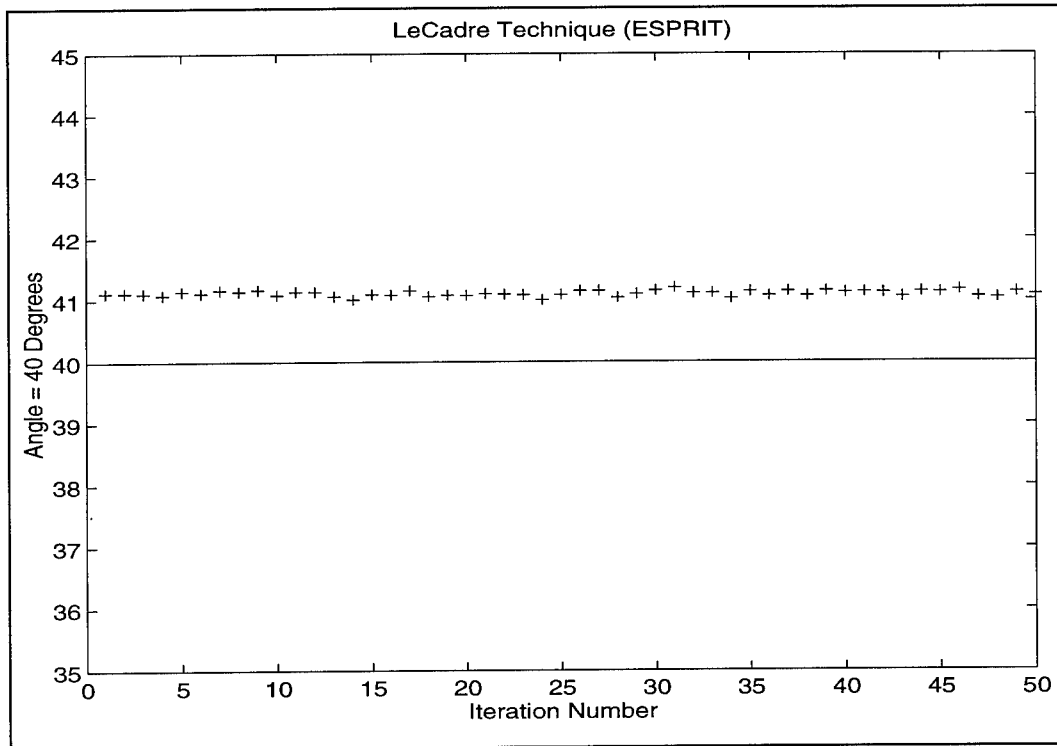
Figure 3-11. <u>Angle = 40° as a function of iteration number</u>
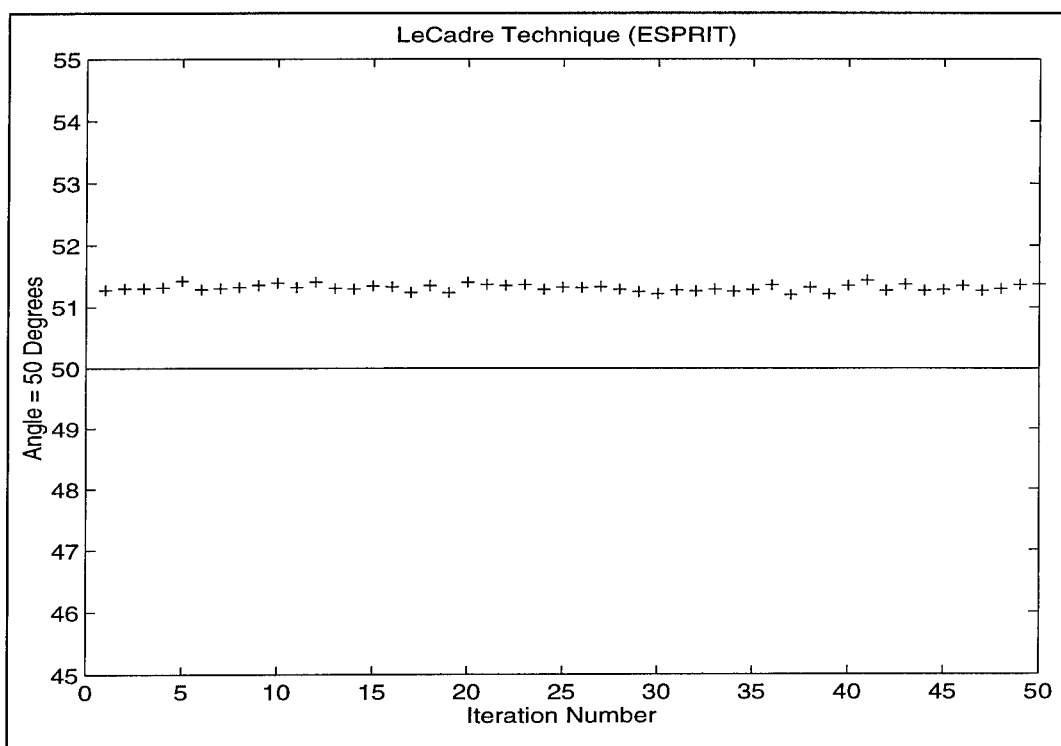
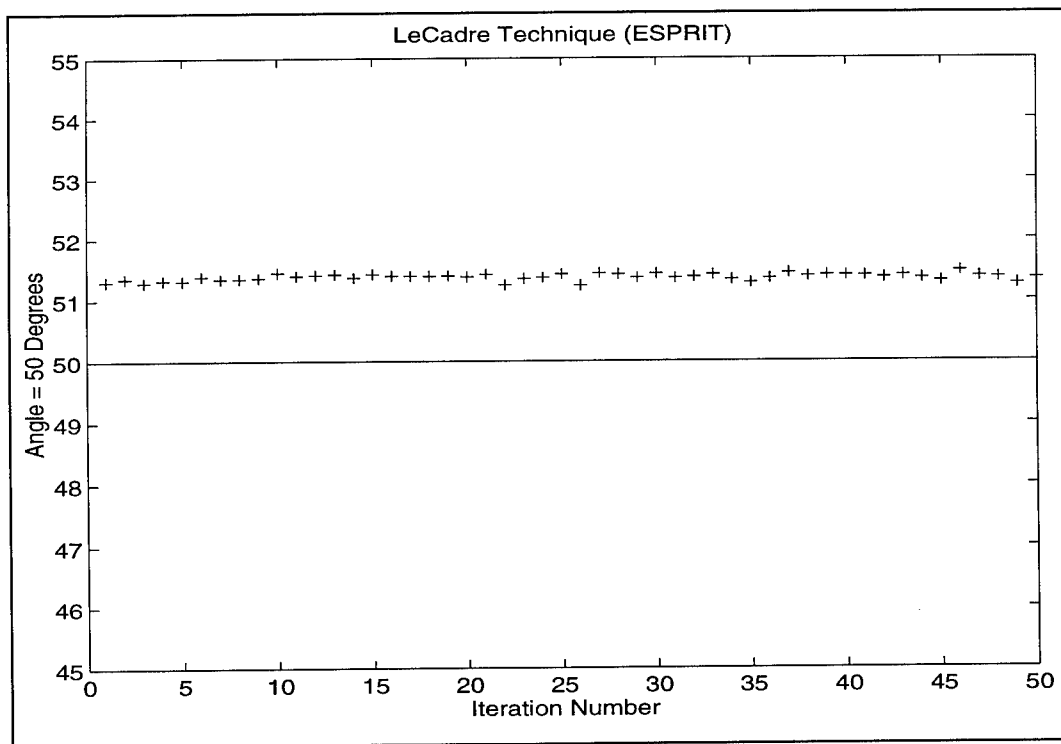
Figure 3-12. <u>Angle = 50° as a function of iteration number</u>
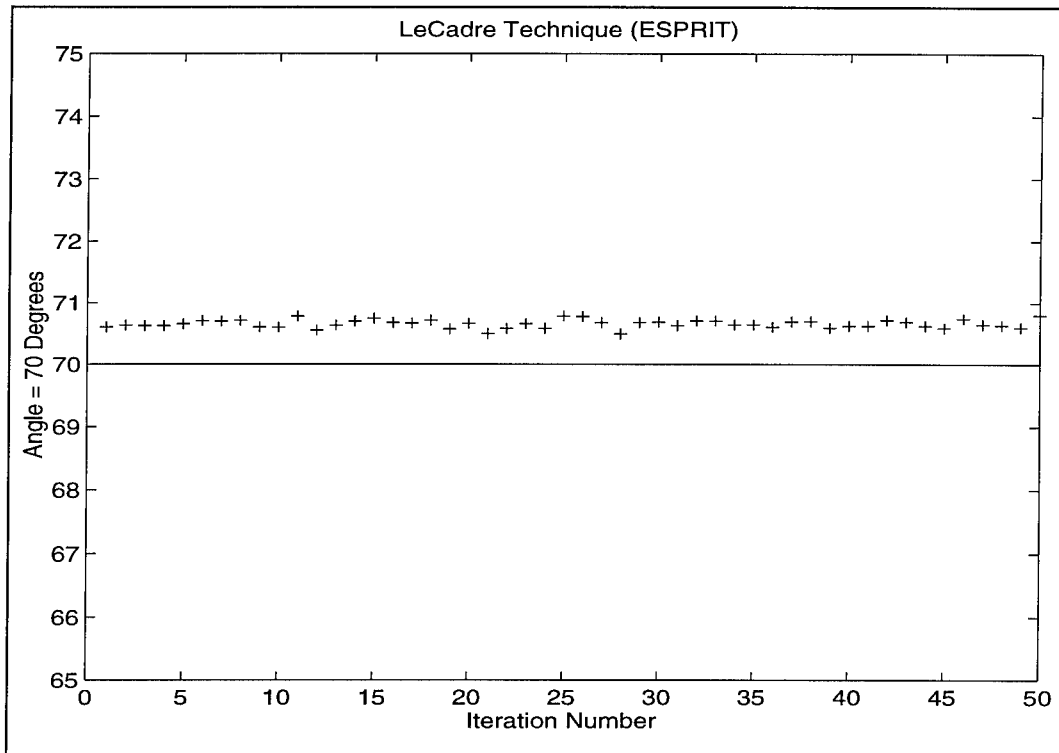
130
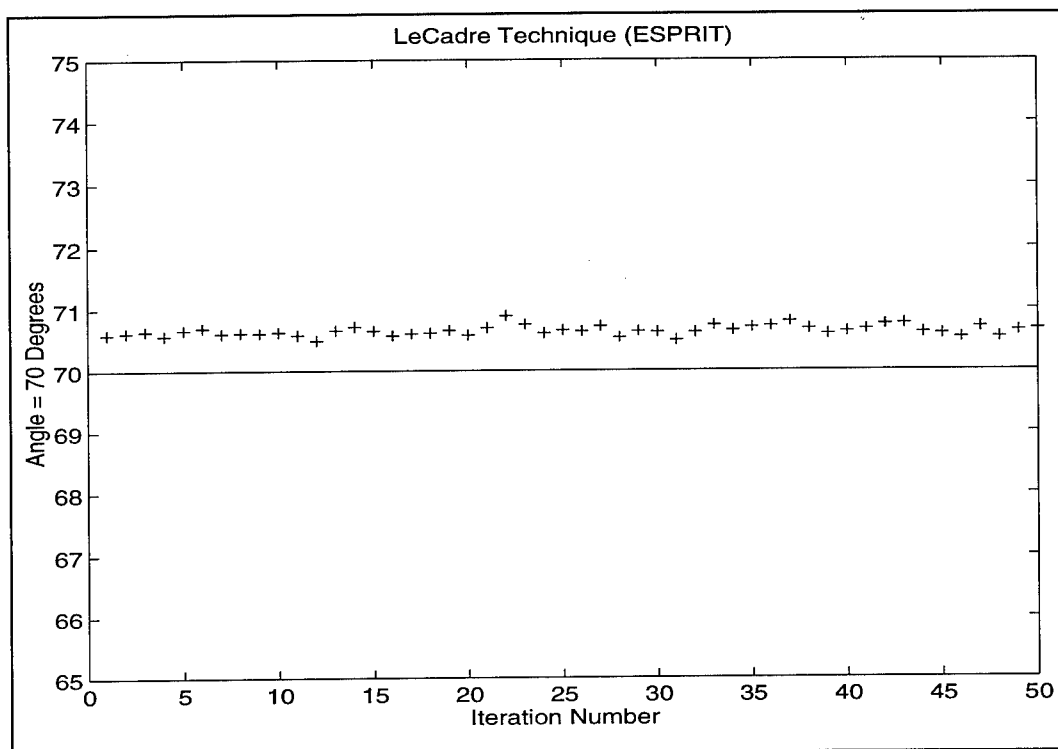
**Figure 3-13. <u>Angle = 70° as a function of iteration number</u>**
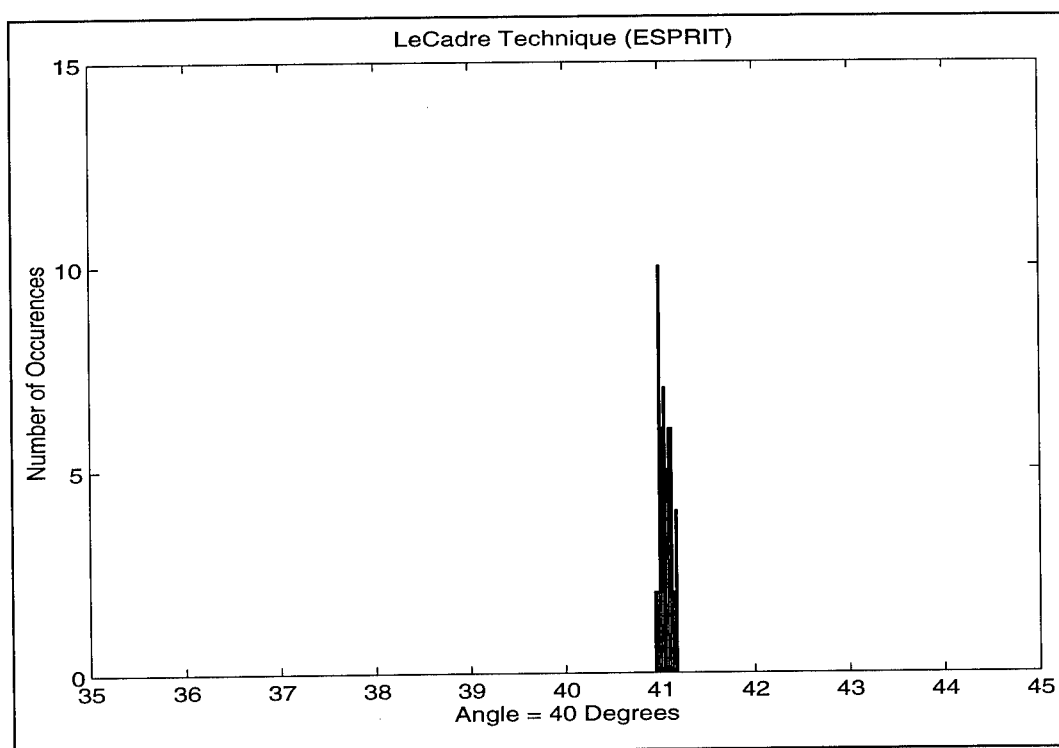
Figure 3-14. <u>Angle = 40° (Histogram)</u>

131
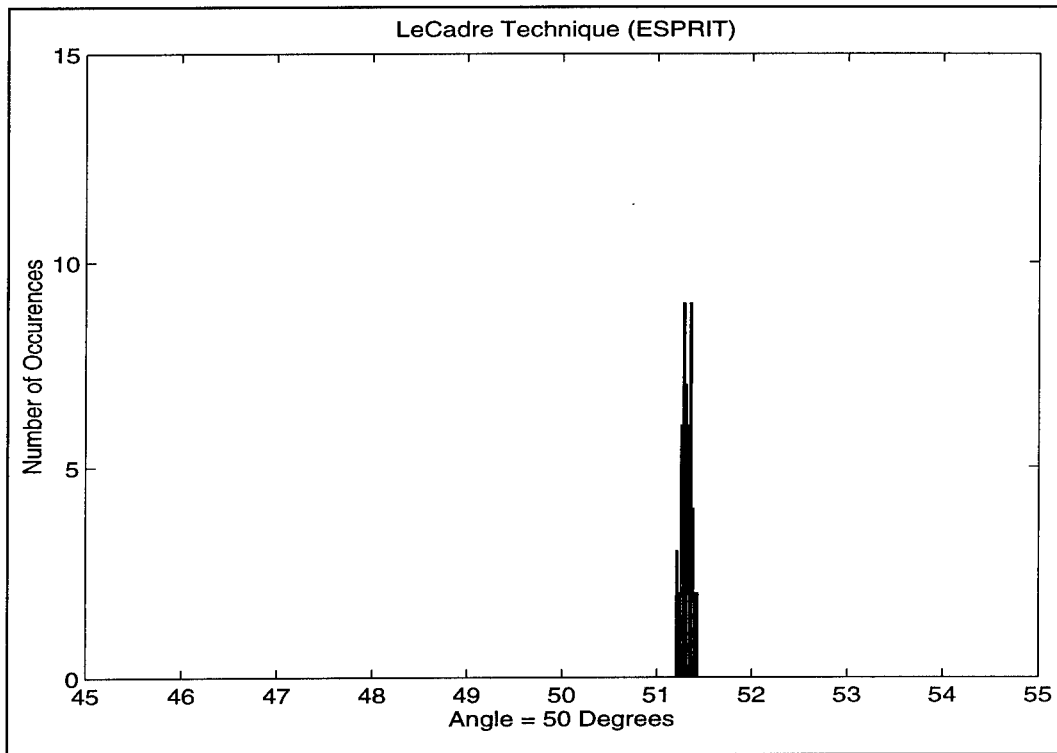
Figure 3-15. <u>Angle = 50° (Histogram)</u>
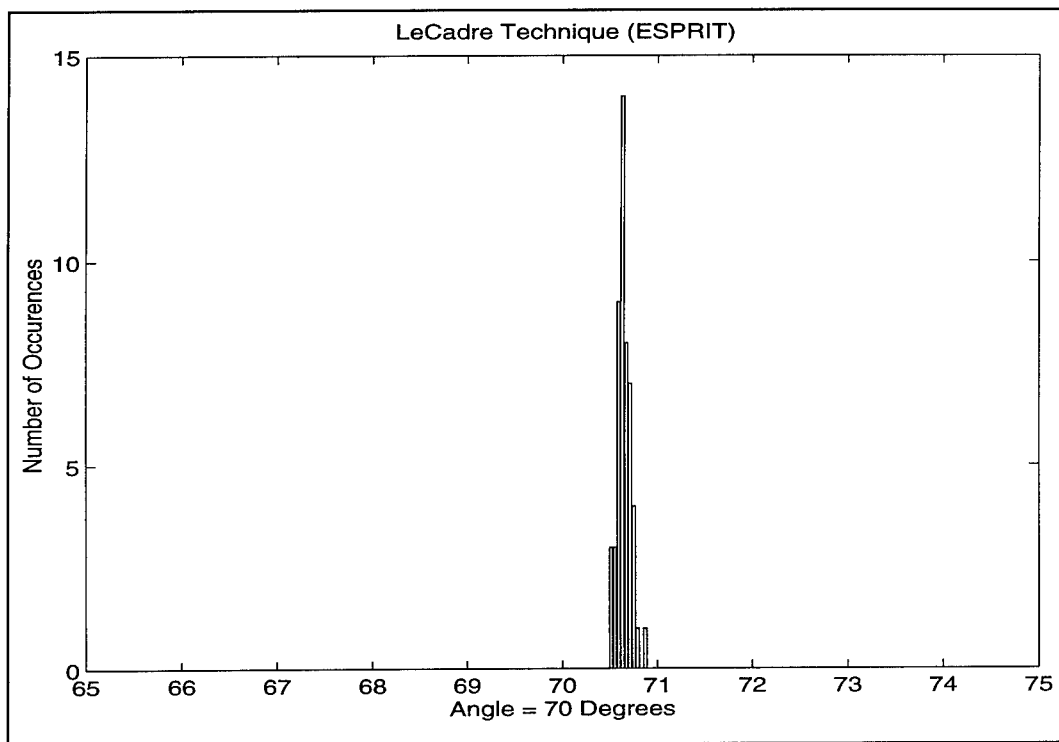
Figure 3-16. <u>Angle = 70° (Histogram)</u>

132

Even-though the angle estimates have a small bias, the variances of these estimates are very small. This clearly shows the effectiveness of the proposed technique.

3.4.2 - Signal Plus Noise Case With Estimated Whitening Functional

In this case, we estimate the whitening likelihood functional using the method derived in section 2. Recall that the first step in this method is to estimate the order of the AR model used to describe the noise and the AR parameters, along with the number of signals. Once these parameters have been estimated, we generate the AR covariance matrix from which the whitening functional is derived. This functional is then applied to the covariance matrix of the received data.

3.4.2.1 - MUSIC Operator

In this section, we ran the same scenario as described previously. However, in this case the whitening functional was estimated whereas in the previous case the true whitening functional was used. The following figure shows the result of the MUSIC operator. Again, the solid line shows the case of whitening case whereas the dotted line shows the un-whitened case.
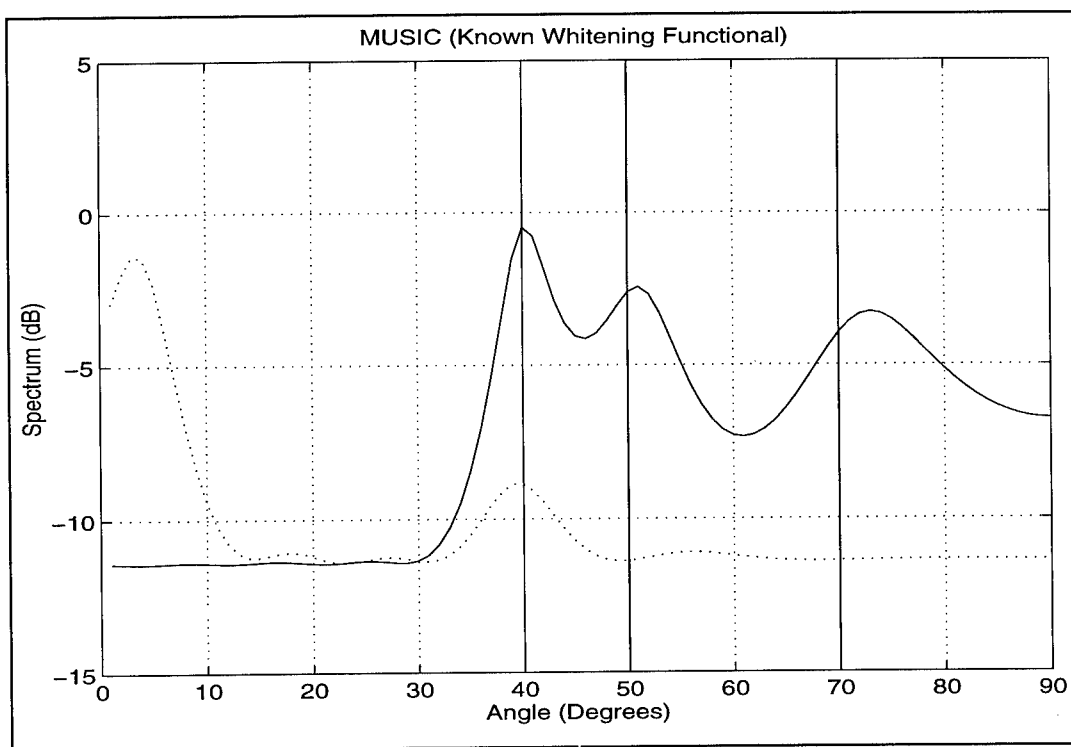
Figure 3-17. MUSIC Algorithm

Note that the results are not as good as in the previous case. However, one can see that after several iterations, the locations of the targets are clearly estimated using MUSIC in conjunction with the whitening functional. Again as in the above case, the solid line shows the results of the whitening process while the dotted line shows the results of the un-whitened case.

3.4.2.2 - ESPRIT Operator

We have selected the same scenario as previously in which we considered an array consisting of 14 sensors spaced at half wavelength and 3 targets located at 40°, 50° and 70°, with an SNR of 5 dB. The AR process was assumed to be of order 2 with parameters 1, -0.9 and 0.2. The following are the results of 50 runs.

	$\theta_1 = 40°$	$\theta_2 = 50°$	$\theta_3 = 70°$
Mean	41.1016	51.3700	70.6565
Variance	0.0018	0.0026	0.0046
MSE	60.6799	93.8408	21.5495

Table 3-4. ESPRIT Estimates

Note again that the bias in the estimates is very small. However, the variances of these estimates are extremely small. Therefore, one can see the advantages of this technique as compared to other techniques such as MUSIC. Note also that these estimates are very good because of the fact the we used an SVD decomposition on the two sub-matrices involved in the matrix pencil. Again as we did before, we can visualize these results by looking at the angle estimates as a function of the number of iterations and the histograms of the estimates.

Figure 3-18. <u>Angle = 40° as a function of iteration number</u>

Figure 3-19. <u>Angle = 50° as a function of iteration number</u>

135

Figure 3-20. <u>Angle = 70° as a function of iteration number</u>
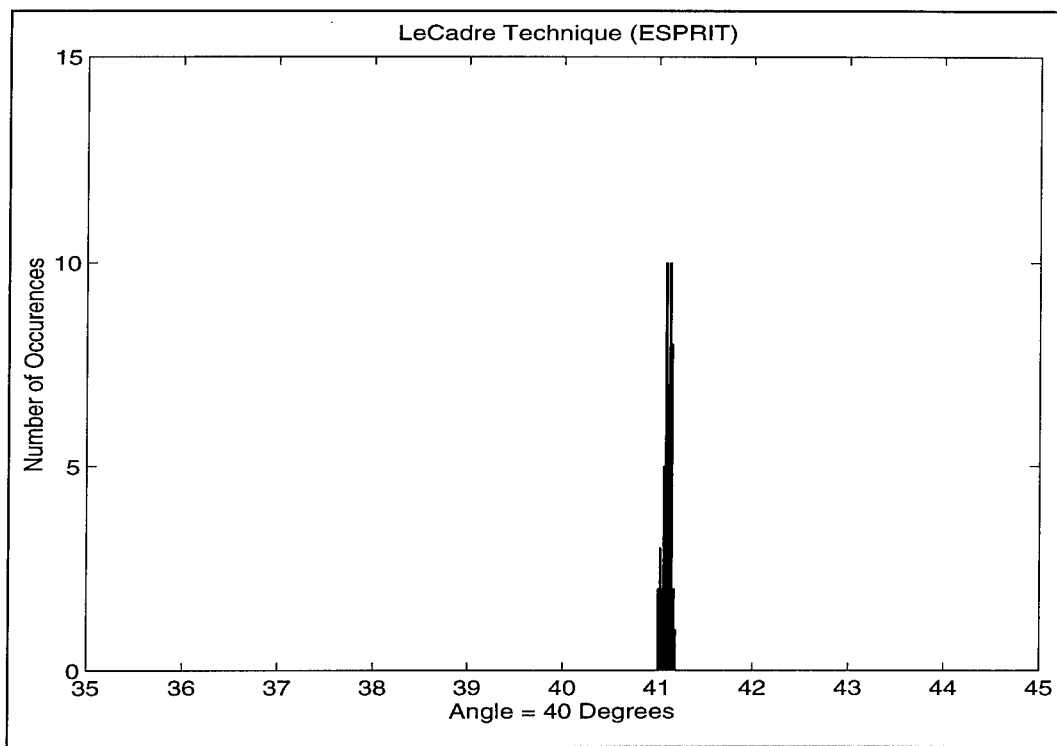
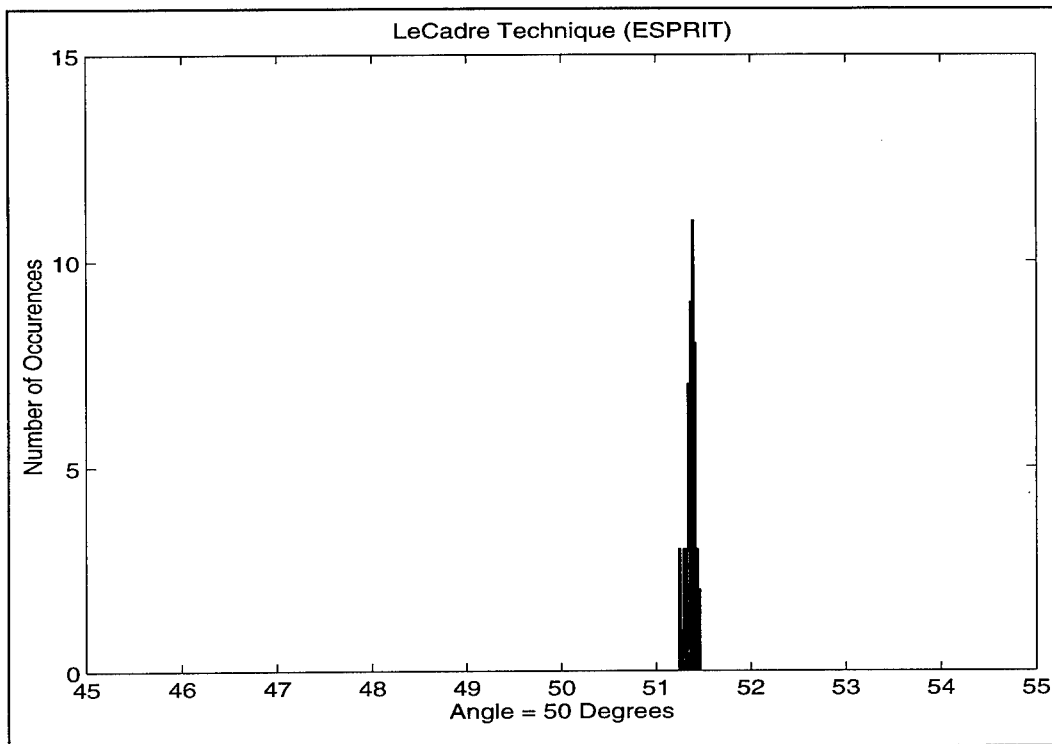**Figure 3-21. <u>Angle = 40° (Histogram)</u>**
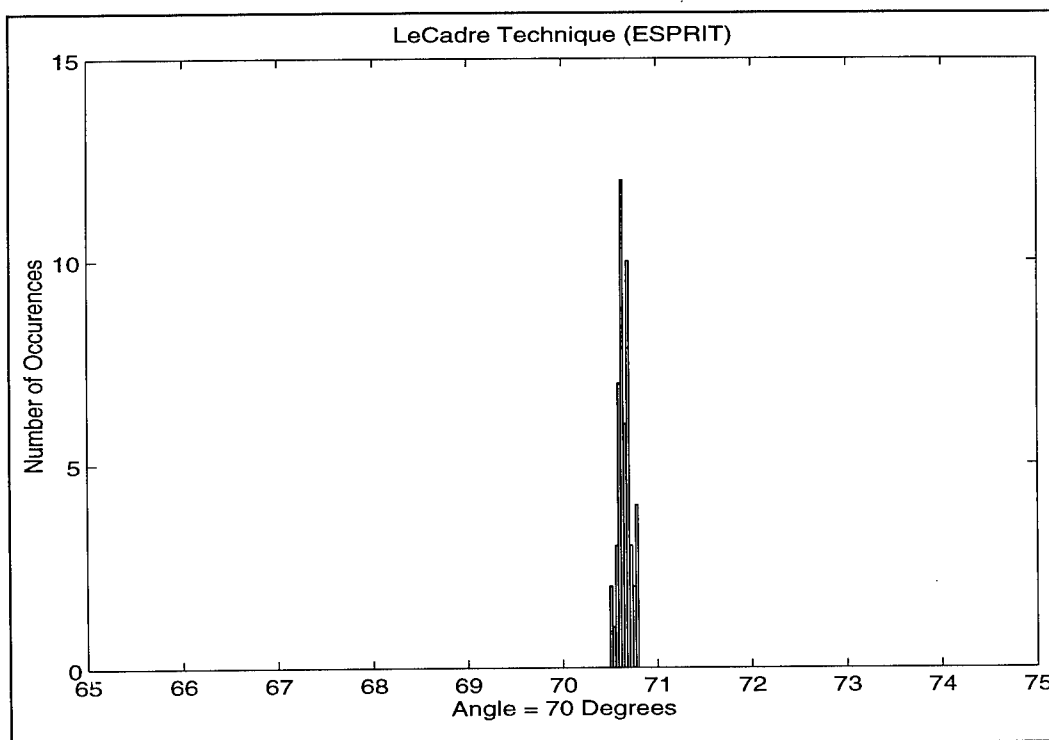
136

Figure 3-22. Angle = 50° (Histogram)

Figure 3-23. Angle = 70° (Histogram)

137

Even-though the angle estimates have a small bias, the variances of these estimates are very small.

3.4.3 - Signal Plus Clutter Plus Noise With Estimated Whitening Functional

In the following figures, the signal was assumed to be of the form

$$\underline{Y} = A\underline{S} + C\underline{N}_1 + \underline{N}_2,$$

where N_1 and N_2 are independent white noise processes. We only consider the case where we estimated the functional and applied it to the AOA algorithms being used. To achieve this, we first estimated the AR parameters and computed the AR covariance matrix from which the whitening functional was obtained. The following figures show the results of the computer simulations. Note that from each case, we first present the values of the AR parameter estimates used in the estimation of the angles of arrival, then the corresponding plots or tables.

3.4.3.1 - MUSIC Operator

In this section we present the results of the MUSIC algorithm.
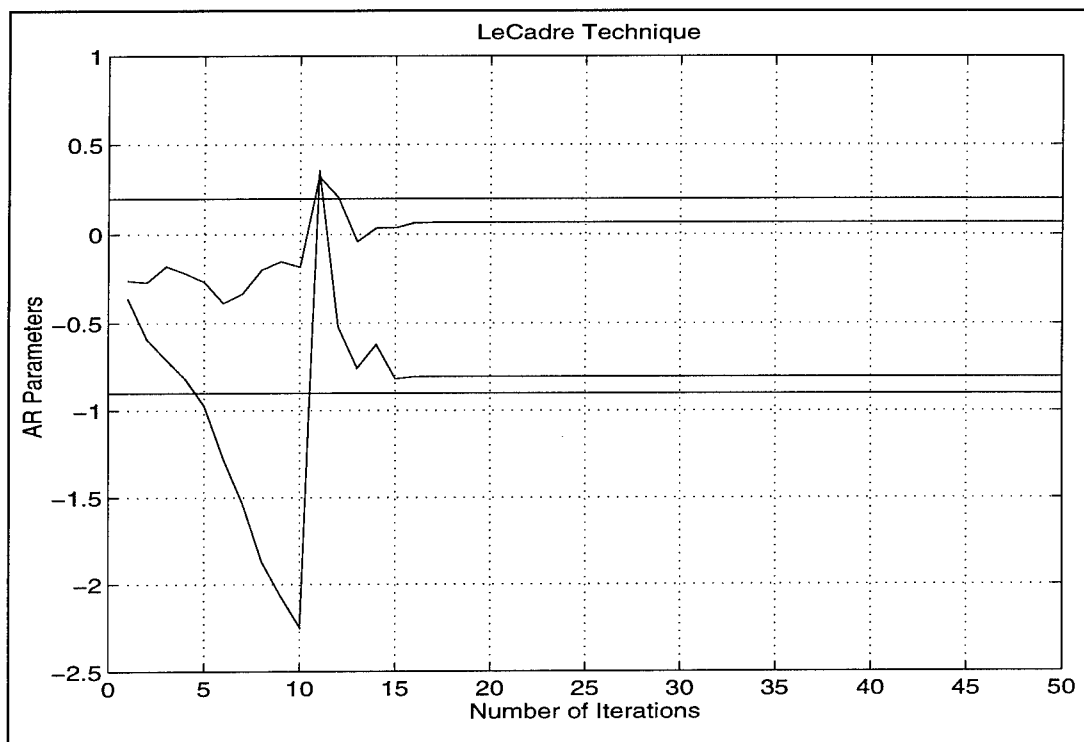
Figure 3-24. AR Parameter Estimates (ρ = 0.6)
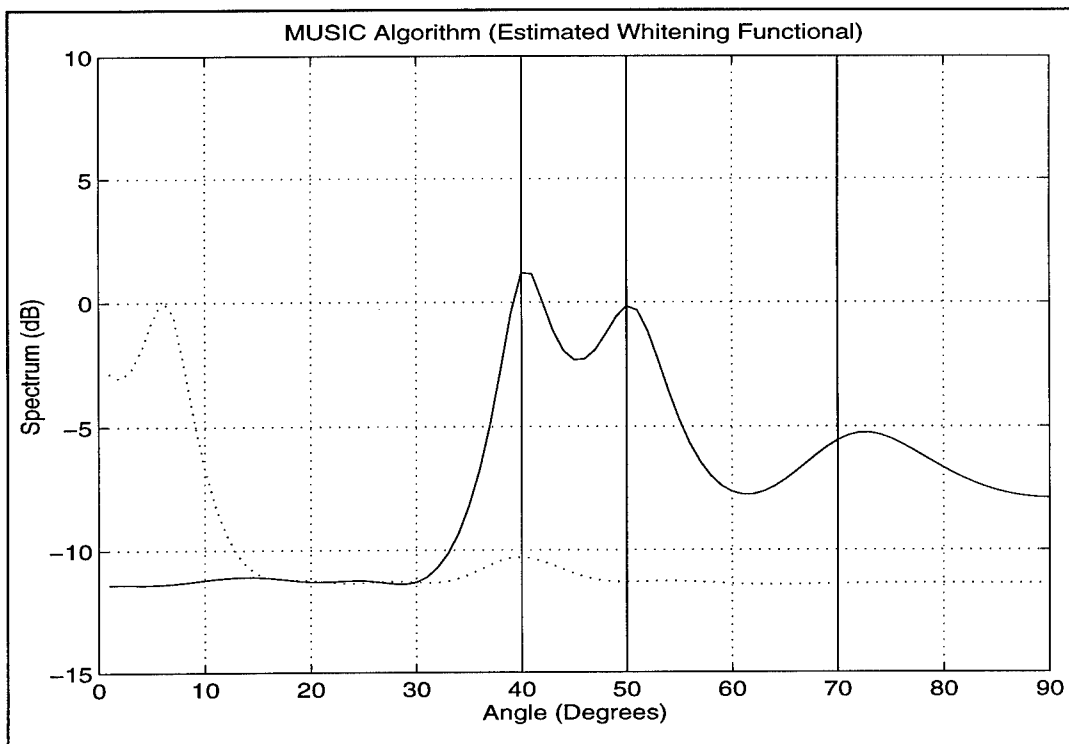
Figure 3-25. MUSIC Algorithm with ρ = 0.6
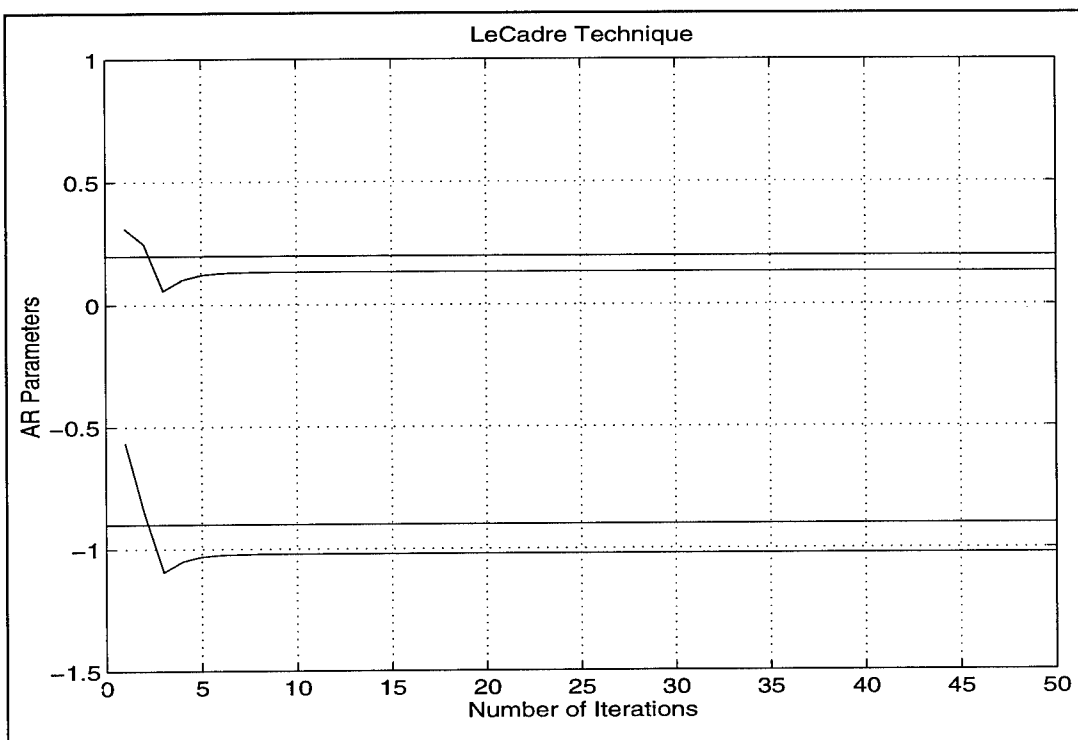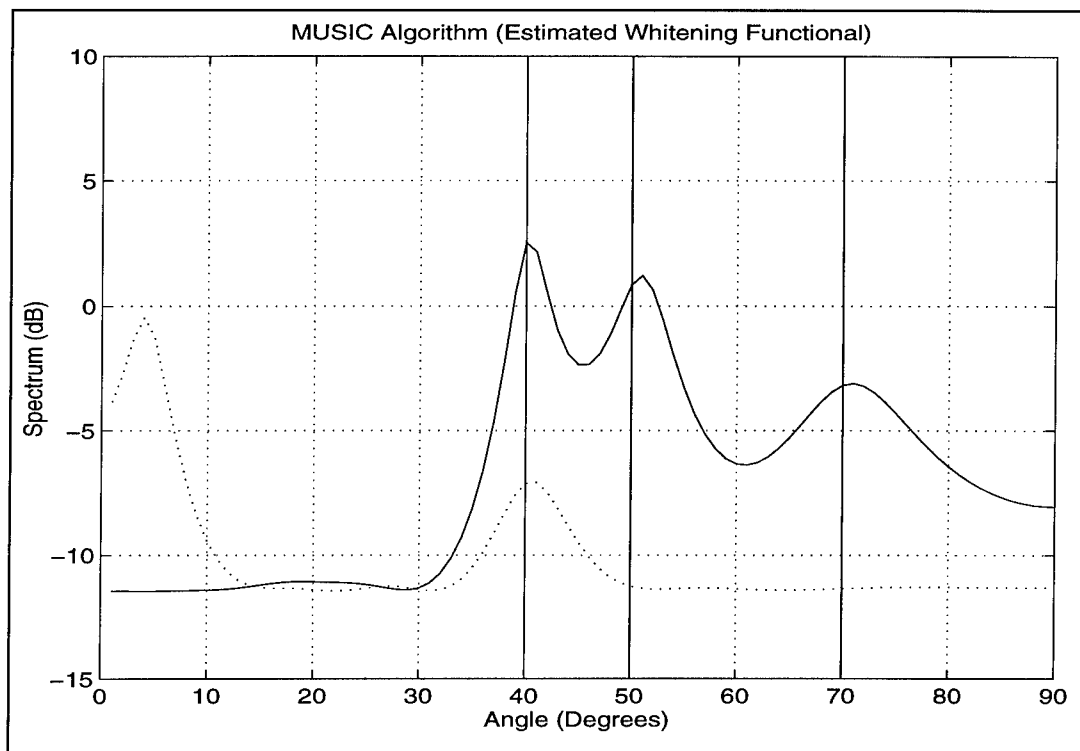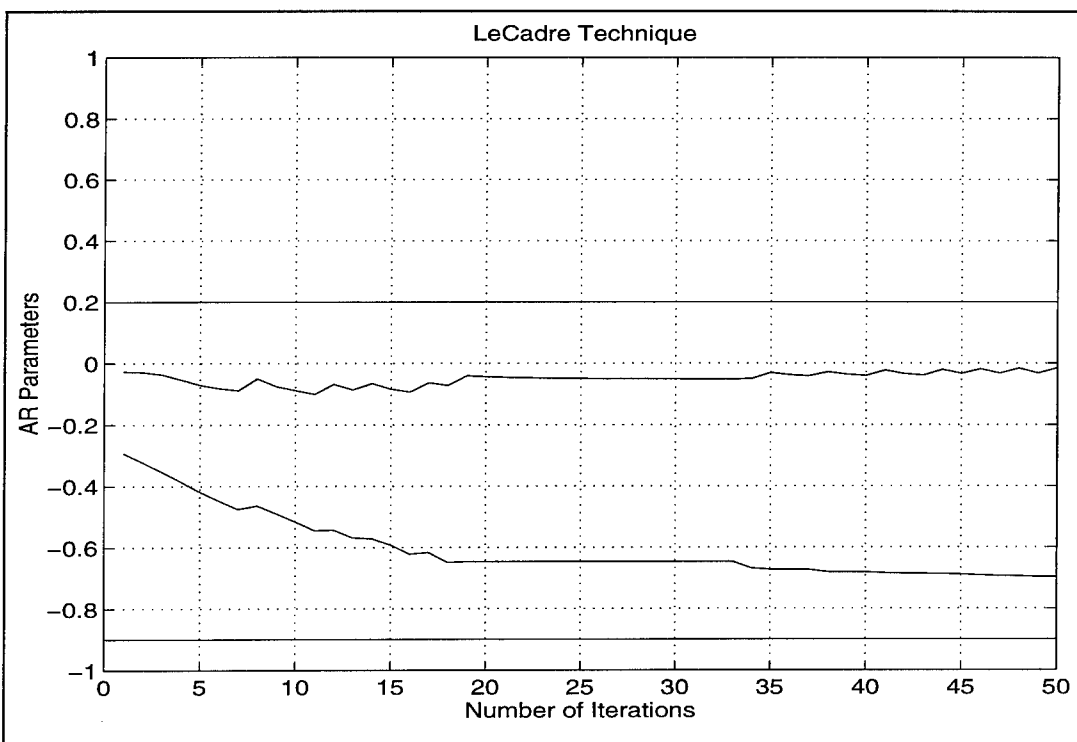

Figure 3-26. AR Parameter Estimates (ρ = 0.7)

139

Figure 3-27. **MUSIC Algorithm with ρ = 0.7**

Figure 3-28. **AR Parameter Estimates (ρ = 0.6)**
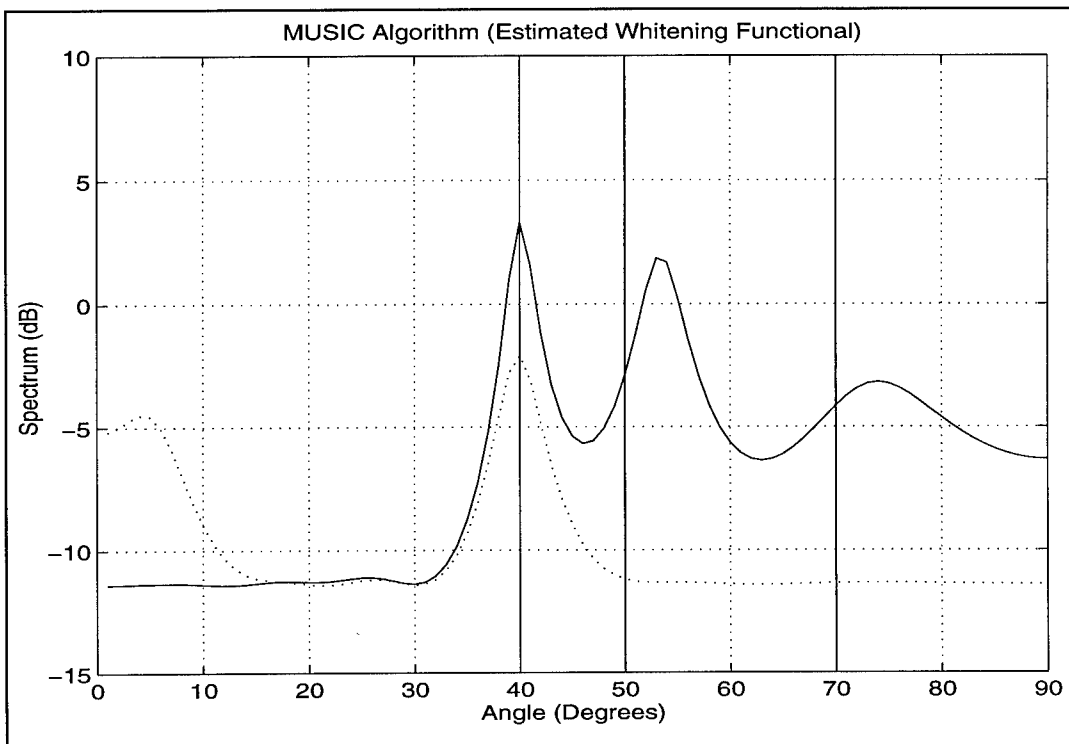
140

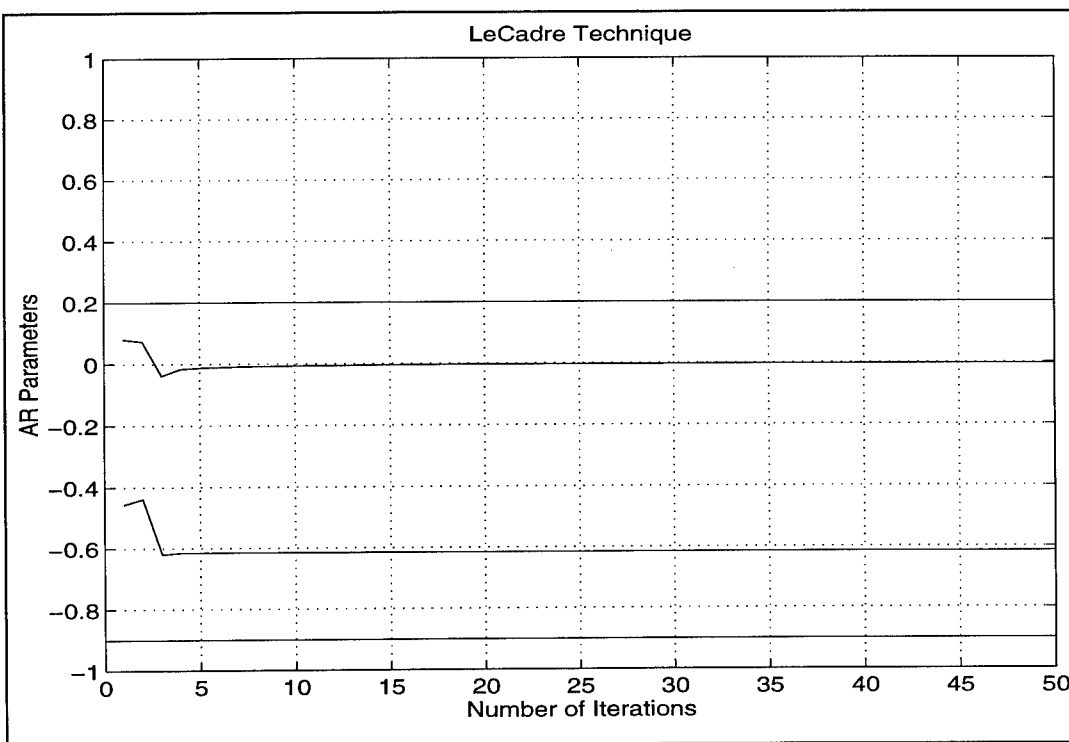**Figure 3-29. MUSIC Algorithm with ρ = 0.6**
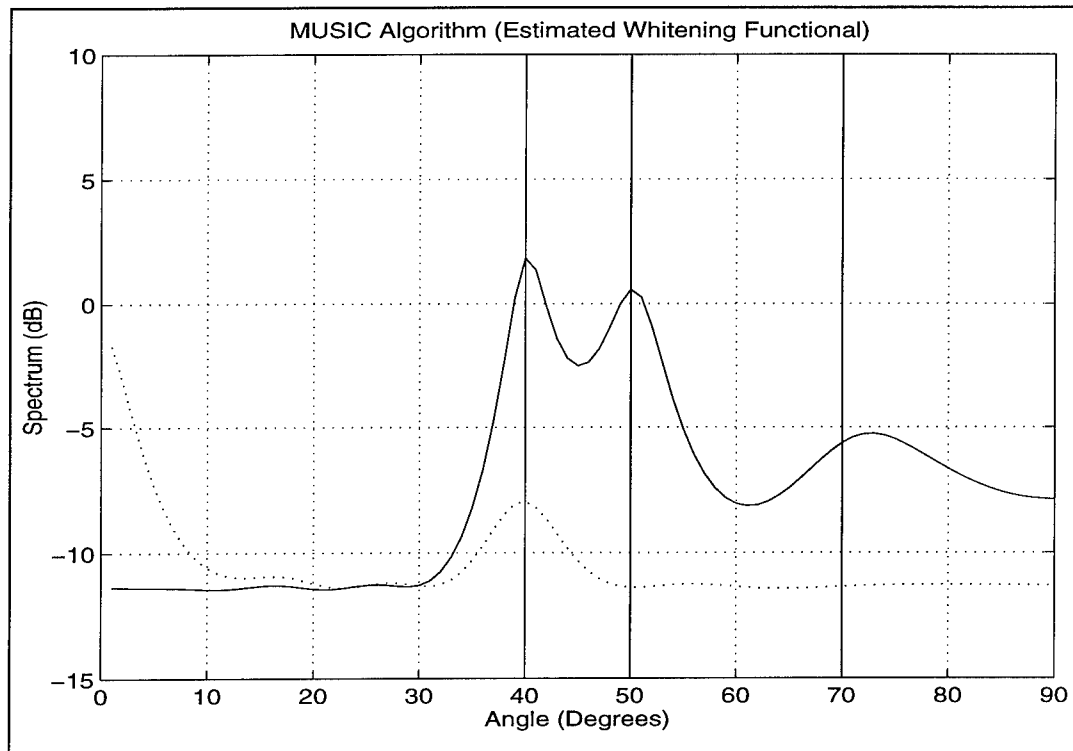

Figure 3-30. AR Parameter Estimates (ρ = 0.7)

141

Figure 3-31. MUSIC Algorithm with ρ = 0.7

Because the AR parameters have been estimated using the technique described above as well as the corresponding covariance matrix from which the whitening functional was derived, one would expect the results to be slightly different than those of the known likelihood functional. To our surprise, the results are not as bad as one would expect as can be seen in Figures (3-24) through (3-31). The locations of the signals were estimated quite accurately, except for small biases. This clearly shows the efficacy of the proposed technique.

3.4.3.2 - ESPRIT Operator

We have selected the same scenario as previously in which we considered an array consisting of 14 sensors spaced at half wavelength and 3 targets located at 40°, 50° and 70°, with an SNR of 0 dB. The AR process was assumed to be of order 2 with parameters 1, -0.9 and 0.2. The following are the results of 50 runs.

	$\theta_1 = 40°$	$\theta_2 = 50°$	$\theta_3 = 70°$
Mean	39.8638	49.9485	69.9631
Variance	0.2689	0.1285	0.2760
MSE	0.9268	0.1327	0.0679

Table 3-5. <u>ESPRIT Estimates</u>

Note again that the bias in the estimates is very small. However, the variances of these estimates are extremely small. Therefore, one can see the advantages of this technique as compared to other techniques. Note also that these estimates are very good because of the fact the we used an SVD decomposition on the two sub-matrices involved in the matrix pencil. Again as we did before, we can visualize these results by looking at the angle estimates as a function of the number of iterations and the histograms of the estimates.
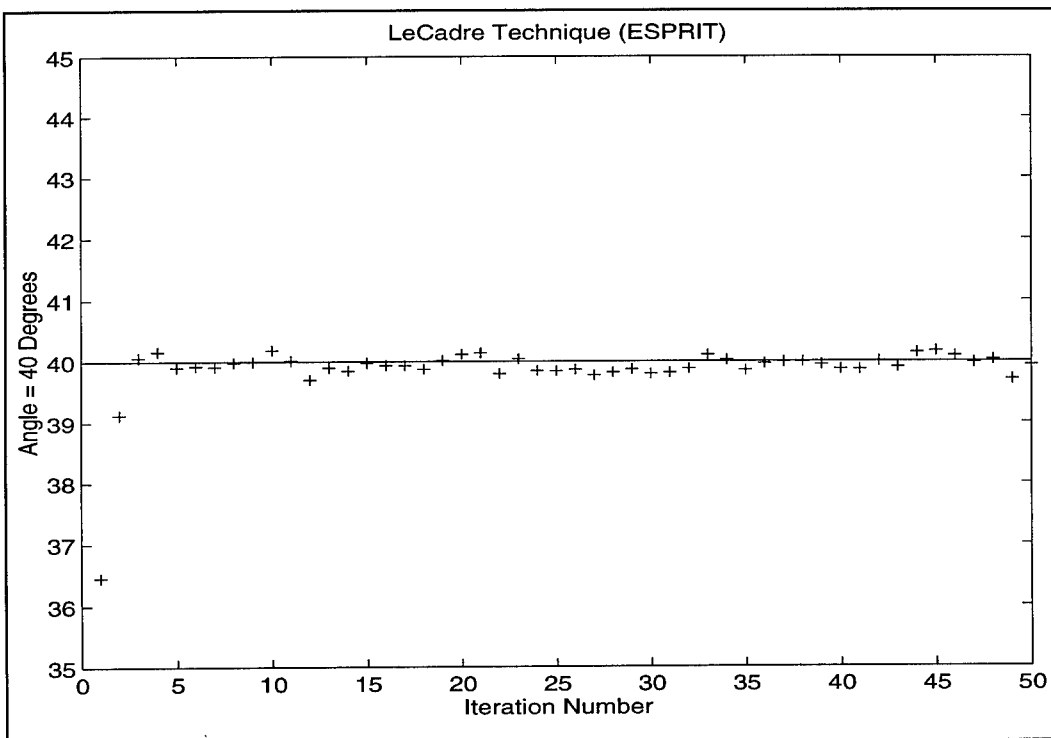

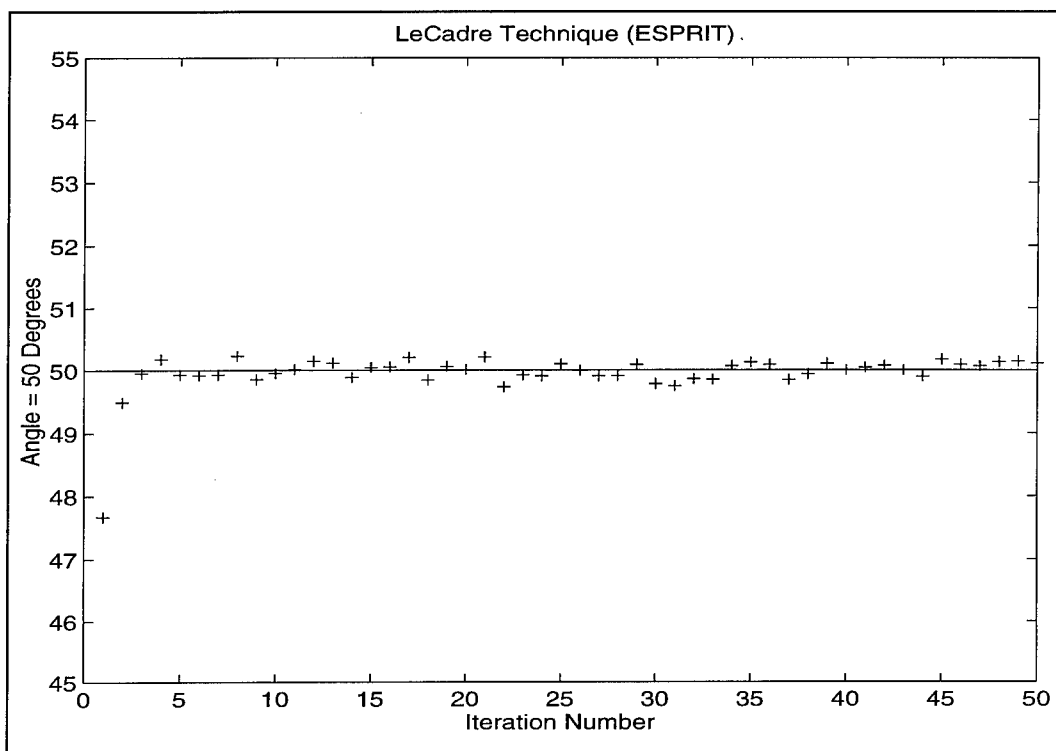**Figure 3-32. <u>Angle = 40° as a function of iteration number</u>**

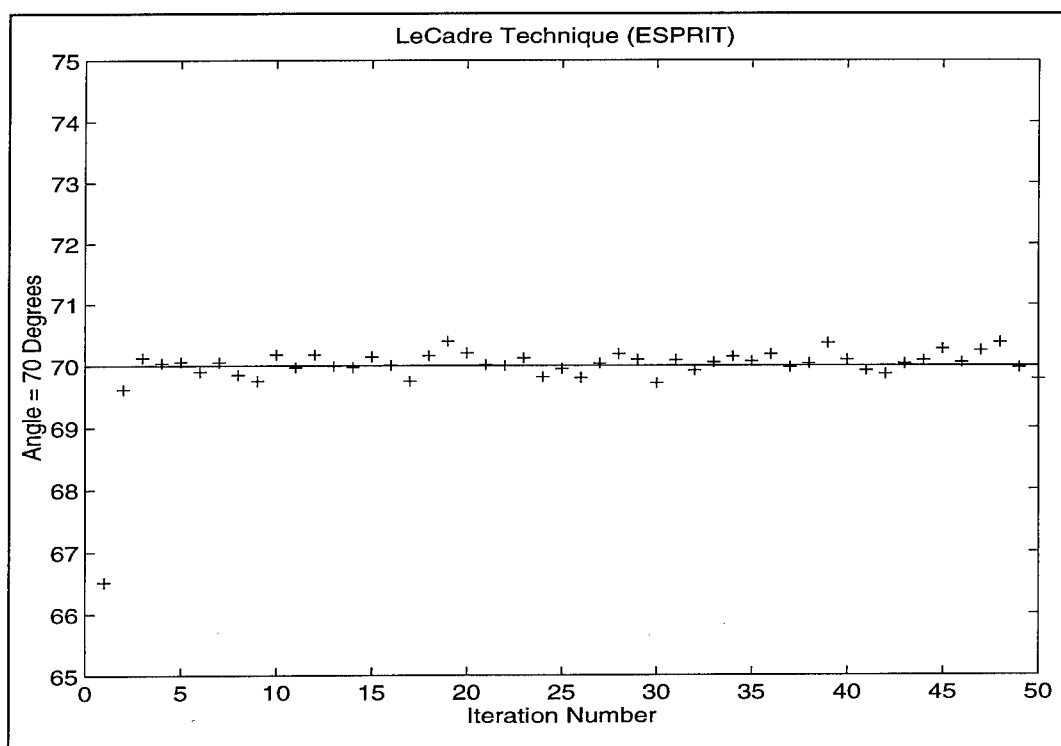**Figure 3-33.** <u>Angle = 50° as a function of iteration number</u>

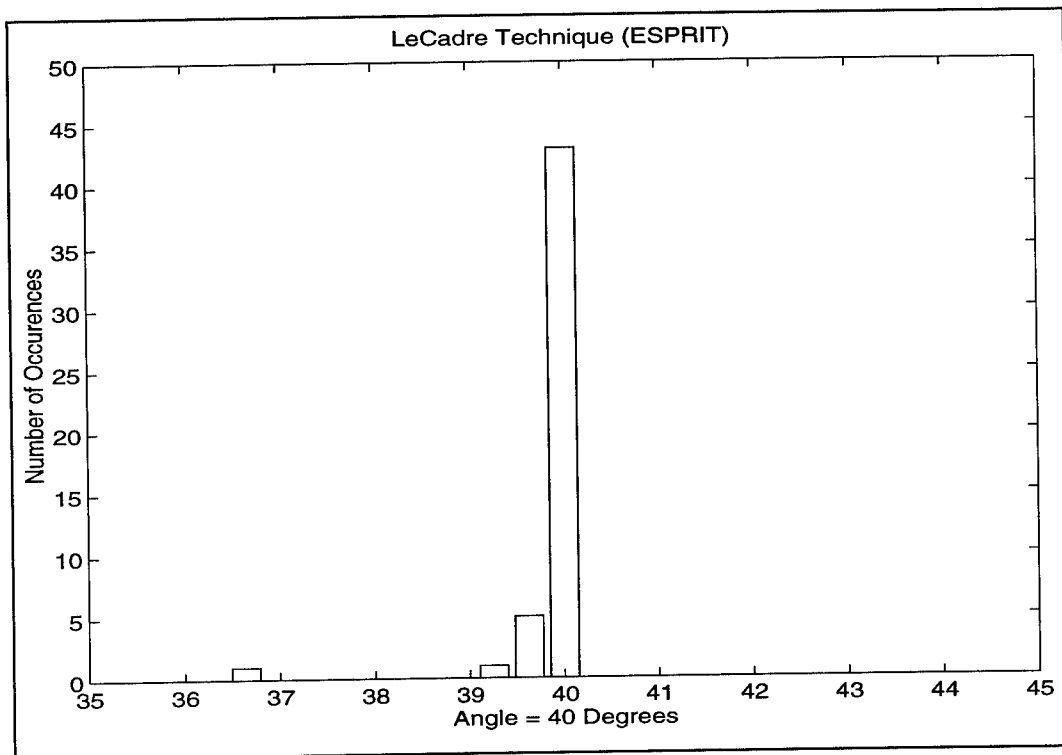**Figure 3-34.** <u>Angle = 70° as a function of iteration number</u>

144

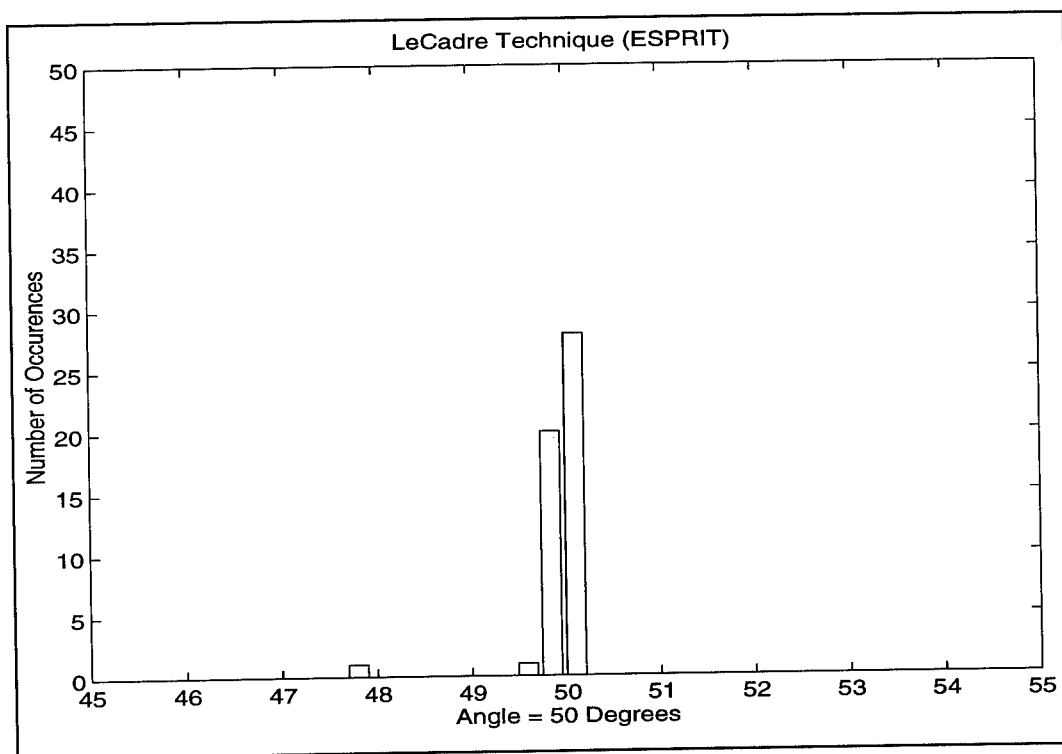**Figure 3-35. Angle = 40° (Histogram)**

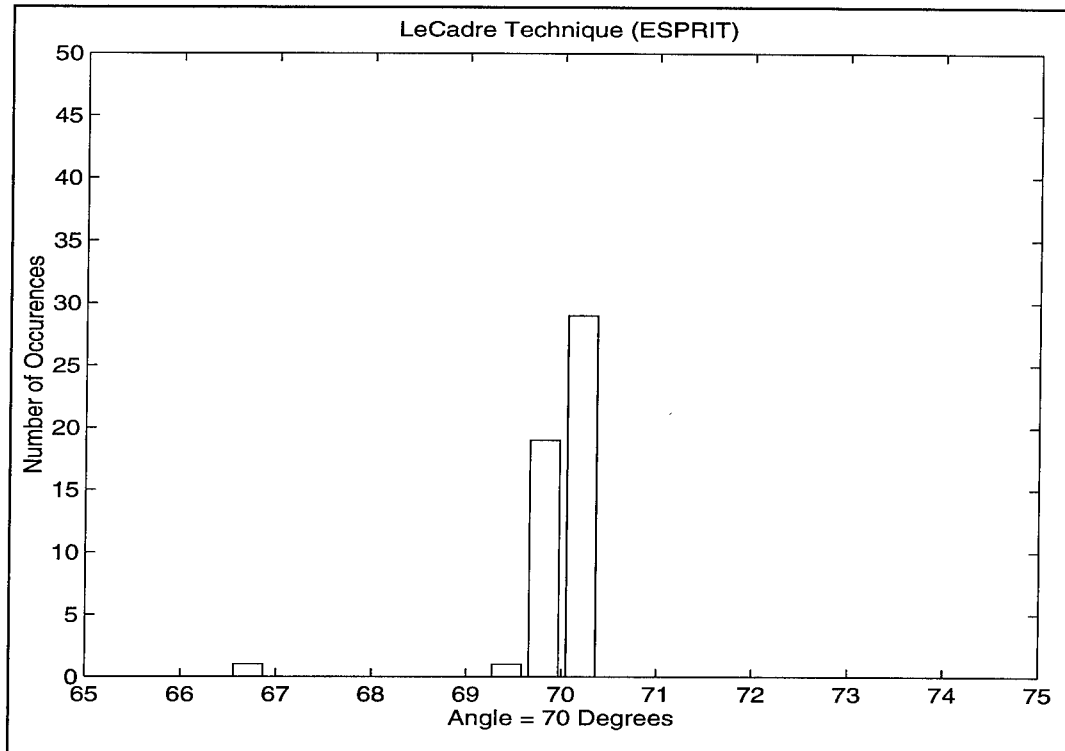**Figure 3-36. Angle = 50° (Histogram)**

Figure 3-37. **Angle = 70° (Histogram)**

Even-though the angle estimates have a small bias, the variances of these estimates are very small

3.5 - Effects of Correlation in the AR coefficients

We then investigated the performance of the technique in estimating the AR parameters. It was observed that the estimation of these parameters depends greatly on the correlation between these parameters. For example, in the case where the parameters are uncorrelated (or very small correlation), the algorithm (minimization of the whitened likelihood functional with respect to the AR parameters) performed very well in the sense good estimates were obtained just after few iteration (10-15) depending on the scenario used. However, in the case where the AR parameters were strongly (or highly) correlated, the algorithm did not perform as well, since over 100 iterations are needed to get good estimates of these parameters. Furthermore, the estimates are biased.

146

Conclusions and Future Recommendations

A new procedure referred to as A'SCAPE has been introduced in the first part of this report. The goal of A'SCAPE is to partition a scene into homogeneous regions and characterize statistically the different regions. As explained in Chapters 2 and 3, two procedures are used to partition the scene. First, a Mapping procedure is used to separate between regions of different average magnitude levels. This procedure is a new adaptive thresholding technique which proves to be very effective at separating regions even when their average magnitude levels are very close to the point that the area of overlapping between their histograms is large. Then, when all regions of different average magnitude levels have been separated, a Statistical procedure is used to separate between sub-regions of different data distributions within each region declared as homogeneous by the Mapping procedure. The Statistical procedure is based on the Ozturk algorithm which needs only 100 samples to properly approximate the PDF of a region. In the Statistical procedure, Gaussian and non-Gaussian regions are formed within each region declared as homogeneous by the Mapping procedure. Outliers are detected which may represent small patches with not enough pixels to be characterized as sub-regions. Also, contiguous non-Gaussian regions that can be approximated by the same PDF are grouped to form the subpatches within each region previously declared as homogeneous by the Mapping procedure. Furthermore, PDFs of the non-Gaussian regions are approximated.

Though using two procedures, A'SCAPE needs four stages to achieve its aim at separating the different contiguous non-homogeneous regions that may exist in a scene. These stages consist of a Preprocessing stage where classical time-frequency processing of the data is performed, a Mapping procedure stage, a Statistical procedure stage, and an Indexing stage which assigns a set of descriptors to every pixel in the scene under investigation. When A'SCAPE is followed by a detection stage, all information is available with respect to which detector should be used for every pixel in the scene.

Interactions between the different stages of A'SCAPE are controlled through an expert system shell, IPUS. This is important because the goal behind A'SCAPE is to monitor an environment which is unknown a priori. Thus, IPUS is needed to ensure that the setting of the control parameters of the different algorithms in A'SCAPE is well suited with the type of data being analyzed. If not, a discrepancy is detected and IPUS searches for the distortion that

may have caused it. This results in the choice of another setting of the control parameters and in the reprocessing of the data.

The A'SCAPE procedure is illustrated through an example of real IR data collected over a lake and land regions. First, the preprocessing stage selects those pixels in the scene that result in uncorrelated data and feeds them to the mapping procedure. Then, the mapping procedure partitions the scene into three main patches: lake, land, and a subpatch within the land. By analyzing the statistics of the three regions it is concluded that the sub-patch within the land may be a body of water. Following that, the statistical procedure partitions further the lake and land into main Gaussian patches and a total of fifteen non-Gaussian subpatches. Also, outliers representing tiny patches, such as roads, have been located and approximating PDFs have been determined for the non-Gaussian subpatches.

A demo package built in Matlab is included which describes in detail the different stages of A'SCAPE. The package has a friendly mouse driven graphical user interface (GUI) and consists of two main sections. The first section presents the detailed steps of the real IR data example. The second section is subdivided into two subsections where in the first one a set of examples is presented which illustrate the need for A'SCAPE, whereas in the second a description is given for every stage of A'SCAPE. The views can be displayed manually or in an automated way as a slide-show. A movie is included which shows the steps the scene goes through during the mapping procedure. This demo package is among the deliverables.

Work under investigation and future work include:

(1) Tailoring/tuning of the edges for subpatches detected by the statistical procedure which is limited by the requirement of 100 reference pixels for the Ozturk algorithm. Note that when the subpatches have more than 100 pixels it is possible to do more processing to fine tune the edges.

(2) Study of the performance improvement of the non-Gaussian detectors over the Gaussian detector in different types of environments and under different circumstances.

(3) development of more expert system rules to enable A'SCAPE to be applied to different types of data (e.g., radar, IR, sonar, medical imaging, etc.).

149

(4) Application of A'SCAPE to medical imaging (e.g., detection of tumors in lung), to law enforcement (e.g., detection of concealed weapons underneath clothing) and to other areas.

(5) Design of a detection stage. As mentioned before, if A'SCAPE is to be followed by a detection stage, the type of detector is selected according to the approximating PDF found for the patch under investigation and parameters for the sufficient statistic of the detector are determined from the pixel descriptors available at the indexing stage. Work needs to be done to test the detection stage, study its performance, and define expert system rules to overcome any possible discrepancies.

(6) Incorporation of A'SCAPE into Rome Laboratory Space-Time Adaptive Processing (RL/STAP) algorithms to enable the characterization of the scene under investigation prior to any detection. As mentioned in this report, when A'SCAPE is done processing the scene all information is available at the indexing stage for every test pixel in the scene with respect to which PDF can approximate the data in the pixel, which patch does it belong to, which pixels can be chosen as reference for the test pixel, etc. All this information is necessary to enable proper choice of the detector type and its corresponding parameters at the detection stage.

The work presented in Part II of this report can be summarized as follows: First, we introduced the problem of detection using non-parametric techniques and showed the disadvantages that are inherent in these techniques. Then we introduced the concept of Parametric detection schemes and demonstrated the need to use such techniques because of the number of degrees of freedom they add. We derived the beamforming algorithm based on LeCadre's work and extended to include both clutter and additive white noise. It is the parameterization of the clutter which gives us the driving factor to include the added degrees of freedom from which the order of the AR(MA) model used is estimated and then estimate the AR(MA) coefficients to obtain the AR(MA) covariance matrix. This matrix will be used to "whiten" the covariance matrix of the received vector. Then the number of desired signals present in the environment is estimated using a modified version of the so-called MDL algorithm. Once these parameters have been estimated, then any direction finding operator

could be used to determine the locations of the signals. Note that only spatial processing has been done at this stage.

In the simulations, we started by checking the convergence of the technique and its ability to estimate correctly the AR(MA) parameters. We found that the technique is very efficient in obtaining good estimates of the AR(MA) parameters with few iterations. Note that this was the case of uncorrelated AR(MA) coefficients. In the case of highly correlated coefficients, as many as 500 iterations are needed to get good estimates of the AR(MA) parameters. Note also that in order to get these estimates, the order of the AR(MA) model has to be known a priori. In most cases, it is estimated from the data set along with the number of signals present in the environment. We actually showed that an extension of the MDL algorithm gave the exact model order and number of signals present.

We then tested the new technique in location estimation. For this, we used two operators, namely the MUSIC operator and the ESPRIT operator. The reasons for choosing these operators is purely arbitrary. Recall that the first operator is called a "search procedure" operator since we have to search over the entire array manifold to obtain the DOA estimates. This may be very computationally extensive and the amount of memory needed to store the results may be very high. For these reasons, we chose an alternative technique, referred to as a "non search procedure" to estimate the DOA's of the signals. It is well known in ESPRIT that the DOA estimates are obtained from the generalized eigenvalues of the a matrix pencil obtained from the received data.

We ran both algorithms and as expected, we obtained excellent results when using the "whitening functional". Failure to do so results in poor estimation of the DOA's. We have even missed some targets as can be seen from the results presented in section 3.

It is our belief that the newly developed technique has great potential in detecting the number of signals as well as determining their locations. It is our recommendation that it be incorporated into the Rome Laboratory Space-Time Adaptive Processing (RLSTAP) capabilities.

In the future, we will include all the temporal effects to the technique to make it a full space-time adaptive technique. It is also our intent to include the effects of mutuals coupling which are inherent when considering an array of sensors. It is our hope and belief that compensation technique will definitely improve the quality of the estimates under study.

MISSION

OF

ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

  a. Conducts vigorous research, development and test programs in all applicable technologies;

  b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

  c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

  d. Promotes transfer of technology to the private sector;

  e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.